SOFTWARE FRAMEWORKS FOR PRODUCTION SCHEDULING AND ANALYTICAL BENCHMARKING

by

ALP EREN AKÇAY

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabanci University
Spring 2008

SOFTWARE FRAMEWORKS FOR PRODUCTION SCHEDULING AND ANALYTICAL BENCHMARKING

APPROVED BY

Assist. Prof. Dr. Gürdal ERTEK ....
(Thesis Supervisor)

Prof. Dr. Gündüz ULUSOY ....

Assoc. Prof. Dr. Bülent ÇATAY ....

Assist. Prof. Dr. Murat KAYA ....

Assist. Prof. Dr. Burçin BOZKAYA ....

DATE OF APPROVAL: ....

Acknowledgments

To start with, I would like to present my deep gratitude and thanks to my thesis advisor Dr. Gürdal Ertek for his endless support throughout my research in the last year of my masters study. His continuous enthusiasm, encouragement and motivation helped me a lot in the preperation of this thesis. Besides the guidance of Dr. Ertek, the completion of this thesis would be impossible without the previous work and efforts of Sergey Drannikov and fruitful collaboration of Tuna Çaprak. I also want to express my thankfulness to the committee members for their helpful reviews, comments and suggestions about the thesis. In addition, my cordial thanks go to all my professors at my masters program. I learned a lot in their courses to strengthen my academic background. My classmates with whom I shared the graduate lab during two years formed a unique intellectual environment for me and I present my best wishes and thanks to them. Lastly, I would like to thank my parents and my brother for their care, love and support throughout my life.

SOFTWARE FRAMEWORKS FOR PRODUCTION SCHEDULING AND ANALYTICAL BENCHMARKING

Alp Eren AKÇAY

Industrial Engineering, Master of Science Thesis, 2008

Thesis Supervisor: Assist. Prof. Dr. Gürdal ERTEK

Keywords: Software Framework Development, Production Planning, Data Envelopment Analysis (DEA)

Abstract

This study is composed of two technical real world problems which are widely encountered in industrial engineering applications. The problems are selected from production scheduling and Data Envelopment Analysis (DEA) which are discussed in two stand-alone chapters. Both of the studies develop practical solutions and software development frameworks to the selected problems. An innovative software accompanies each study in which it is tested and implemented in real world projects. In the first study, a general software framework that integrates optimization into daily production planning in chemicals industry is discussed. The number of required washes during changeovers in a single mixer is minimized through effective production sequencing which converts each production setting into a single Traveling Salesman Problem (TSP). The application of the software at a hygiene products manufacturer yielded significant saving in the consumption of clean water. In the second study, DEA results are considered with a data mining perspective. In order to integrate DEA results with data mining and information visualization techniques, a formal representation of DEA results is proposed in a framework.

ÜRETİM ÇİZELGELEME VE ANALİTİK KIYASLAMALAR İÇİN YAZILIM ÇATILARI

Alp Eren AKÇAY

Endüstri Mühendisliği, Yüksek Lisans Tezi, 2008

Tez Danışmanı: Yrd. Doç. Dr. Gürdal ERTEK

Anahtar Kelimeler: Yazılım Çatı Yapıları Geliştirme, Üretim Planlama, Veri Zarflama Analizi (VZA)

Özet

Bu tezde endüstri mühendisliğinin iki temel alanına yönelik çalışmalar sunulmaktadır. Üretim çizelgeleme ve Veri Zarflama Analizi (VZA) başlıklarından seçilen konular, iki ayrı bölümde anlatılmaktadır. Bu seçilen konulara her iki çalışmada da pratik çözümler ve yazılım çatıları önerilmektedir. İlk çalışmada, eniyileme ve kimya sanayinde üretim planlamayı birleştiren genel bir yazılım çatısı önerilmektedir. Her bir farklı üretim durumunun tek bir Gezgin Satıcı Problemi'ne çevrilerek çözülmesiyle elde edilen etkili bir üretim sırası ile, tek bir karıştırıcının ürün değişimleri sırasındaki temiz su tüketiminin en aza indirgenmesi hedeflenmektedir. Sistemin temizlik ürünleri üreten bir fabrikada uygulanmasıyla önemli miktarda su tasarrufu yapılmıştır. İkinci çalışmada, VZA çözümlerine veri madenciliği perspektifinden bakılmıştır. VZA çözümleri ile veri madenciliği ve görselleştirme tekniklerini bütünleştirebilmek için, bir yazılım çatısı altında VZA çözümlerinin matematiksel bir gösterimi sunulmaktadır.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Two independent topics are discussed in this thesis which aims to achieve the common objective in both problems: improving the existing performance of a system with proposed methods and solutions. The topics are selected from two distinct fields, namely, production scheduling and Data Envelopment Analysis (DEA). The problems are chosen especially in the real world cases that practitioners encounter in the planning of their operations. The commonality of the two studies is the presentation of new software frameworks, accompanied with innovative software tools that implement each framework. The main research question at the heart of each problem is "Which characteristics should a software carry in order to solve such a problem effectively?" Answers are seeked to this question and software frameworks are proposed for each problem. In accordance with the framework built for each problem, a software is designed and developed. Each software is implemented in a real world application.

The two stand-alone problems which are seperated in two distinct chapters are as follows:

- **Chapter 2:** A Decision Support Framework and an Open Source Software for Saving Water in Chemicals Industry

- **Chapter 3:** A Software Framework for the Integration of Data Envelopment Analysis and Data Mining

These two chapters are originally formed as seperate academic articles. Thus, each chapter has its own introduction, literature survey, methodology, results, discussion and conclusion. In this thesis, the content, language and writing style of the papers are kept unchanged but they are rewritten in the required thesis format.

Chapter 2 proposes a decision support framework and an open source software for production planning in chemicals industry to reduce clean water consumption. The source of the research problem is the local branch of a global leader of cleaning products, JohnsonDiversey Turkey. The company is specialized on the production of inustrial hygiene and cleaning products. Liquid products constitute the major part of the product families and they are processed in large mixers. The main problem in the company is the high volume of clean water use during the changeovers in a mixer. Since the company use a single mixer for the production of a special type of product family, a wash procedure is required in order to eliminate the contaminating effects of the processing product on the next one. It is clear that an effective sequencing of products that are assigned to the mixer can decrease the number of required washes on the mixer. In addition to the reduction of the volume of water used in the washes, idle time of the mixer during the washes and labour cost (i.e. the mixer operator carries out the washing procedure) can also be cut down. After examined the production practices of the company, following question is tried to be answered: "Is there a common approach or general framework for production planning in chemicals industry where the mixer cleaning poses high operatonal costs?"

As an answer to this question, a stuctured decision support framework is formed together with an effective and user-friendly software. The framework suggests how to sequence products on a single mixer to get the most effective product sequence to save water during changeovers. In this study, a set of important capabilities such as defining product groups, handling urgent orders and visualizing production schedules are integrated.

The sequencing problem in production devices such as mixers, tools, processing machines etc. is a typical problem type that is widely encountered in process industries like chemicals, beverages, food and pharmaceuticals. Generation of the best possible production sequence can yield not only monetary savings, but also environmental benefits. For instance, the proposed decision support system has realized the saving of an important amount of clean water in JohnsonDiversey Turkey. The presented decision support framework can be easily adopted to another company with similar production practices. Following the main guidelines of the framework, the software can be modified to satisfy the practical requirements and implement specific applications belonging to another company.

Chapter 3 is concerned with the development of a formal software framework for the representation of the DEA results. It is envisioned that such a structured and formal representation will enable analysts to better integrate the data mining and information visualization techniques with the DEA results. In addition to the framework, an innovative DEA solver software, SmartDEA, is designed and developed. The user-friendly software is capable of solving four types of classical DEA models without a restriction on the size of the model unless the number of entities to be analyzed and the number of inputs and outputs of the model are extremely high. It is expected that the development of such a DEA solver, SmartDEA, will lead to convenience for effective solution generation not only in this study but also in prospective studies about DEA. How the solver is used and the structure of the DEA solution data representation are discussed in detail. The test implementation of the software is fulfilled for the spare parts warehouse of a leading manufacturer in Turkish automotive industry. Performances of the dealers are evaluated in terms of the dealers' and the affiliated company's perspectives.

DEA model solutions consist of a set of information about the efficiency scores of the benchmarked entities, their reference sets and projection values for each input and output that are included in the model. The meanings of these terms are discussed in detailed in both theoretical and practical point of views. In accordance with the managerial objectives of the DEA, these results should be examined in the most effective and comprehensive way (1) to reveal the hidden structures existing in the results, (2) to identify important patterns, and (3) to derive executive insights and guidelines regarding the benchmarked entities. In order to do so, Chapter 3 provides a structured framework for the representation of the results data. This will be a basis for further analytical benchmarking activities of DEA results using data mining and information visualization techniques. SmartDEA is developed as a tool to integrate the DEA results and these tecniques by means of the proposed framework. The results and insights that come up with the this integrated approach are very effective tools to develop competitive strategies for the benchmarked entities.

As noticed in the subject and the content of the studies, all of them intend to solve real world problems by developing software frameworks, as a part of applied projects mostly in operational and tactical levels. The given order of the studies in the thesis does not imply any succesive relation between them and reader can follow the chapters independently. The references used in each study are given at the end of the thesis as a whole Bibliography section, but with subsections specifying chapters. If available, the Appendix section of each chapter is affixed at the end of the corresponding chapter before the next study to keep the integrity of seperate chapters.

Chapter 2

A Decision Support Framework and an Open Source Software for Saving Water in Chemicals Industry

In this paper, we describe a general software framework that integrates optimization into daily production planning that is widely encountered in chemicals industry. We also describe how we have realized the framework as a decision support system which solves the product sequencing problem in chemicals processing industry. In chemical industry, washes at chemical processing mixers are typically required to prevent the contaminating effects of the previous products on the current product being mixed. The minimization of the number of required washes is possible through effective production sequencing, yielding a significant saving in the consumption of clean water. As a case study, we implemented and tested our software at a chemical plant located in an industrialized region of Turkey. The plant produces industrial cleaning chemicals and is owned and operated by JohnsonDiversey Turkey, which produces a variety of products for local and global markets. The system reads critical production-related information from a database and generates approximately optimal or at least "good" solutions sequencing the products to be produced at a mixer. The former solution is obtained for small problem instances through solving a Mixed Integer Programming (MIP) model with approximated cost structure, whereas a "good" solution is seeked for larger instances. Our study integrates a unique set of critical features for the first time in literature such as defining product groups, handling urgent orders and effectively visualizing production schedules. The company reported to obtain substantial water savings since this new system was into practice by the production planners.

Software Availability

Name: SmartScheduler

Developers: Alp Eren Akcay, Gurdal Ertek, Sergey Drannikov

Contact address: Sabanci University
Faculty of Engineering and Natural Sciences
Orhanli, Tuzla, 34956, Istanbul - Turkey
Tel.: +90 (216) 483-9568
Fax: +90 (216) 483-9550
ertekg@sabanciuniv.edu

First available: September 2006

Minimum hardware requirements: Intel Pentium 4, 512 MB RAM

Software requirements: Microsoft.NET Framework, Operating system Win 2000, NT, XP

Programming Language: C#

Form of Repository: Microsoft Access database

Language: English, Turkish

Size: 5.34 MB

Cost: Freely available application with open source code and sample data

Software/documentation availability: http://www.opendecisions.org

2.1 Introduction

The performance of operations within a company strongly depends on the quality and success of operational decisions. While reaching such decisions, an important issue in contemporary production systems is taking into account the environmental impacts of production practices and minimizing undesired impacts through technological advancements such as latest production techniques and planning software tools.

This paper describes a framework and an open source decision support system (DSS) for saving water in chemicals industry through effective product sequencing. In chemicals industry, the effective use of water turns out to be an important performance criteria in terms of resource management and environmental perspectives. Considering the large public attention to the consequences of global warming in the last decade, better management of scarce water resources emerges as a top priority for environmental researchers and practitioners in various industries. Use of operational research methodologies to limit natural resources consumption and developing decision support tools as building blocks of production planning software help the manufacturers improve their production practices with respect to the environment.

In chemicals industry, the order of products to be processed in a mixer should be chosen carefully to reduce the amount of water consumed to wash the mixer. For instance, any of the previously processed products can have contaminating effects on the product currently being mixed. In order to keep the chemical composition of the current product unadulterated, the mixer operators check whether a mixer wash is required by considering the ingredients of the products which are processed since the previous wash on the mixer. By ordering the products in a way to minimize washes, operators can significantly decrease the water consumption yielding critical operational cost reductions and environmental benefits. In practice, the need for a quick and reliable planning tool arises especially when the number of products is large and special requirements for the production are present such as processing of urgent orders or grouping of products that belong to the same customer.

In our study, we developed a decision support software that suggests an optimal or "good" product sequence for a single mixer, depending on the problem size and the presence of product groups. In the former case, the optimal solution is found under the assumption of an approximated cost function. Two solution approaches are implemented in the software: For small problem instances, the optimal solution is computed by constructing a Mixed Integer Programming (MIP) model and solving it. For large problem instances, a simulated annealing algorithm is run to come up with a "good" solution in a reasonable time frame (a few seconds). Our solution methodology and decision support system exhibit five distinctive aspects:

1. The system allows forming of product groups in the production list. Production list refers to the set of products that must be processed in a particular time interval (eg. a week). The order of products within a group can be either fixed by the user, or may be changed by the software to minimize water consumption, hence the name flexible.

2. It permits modifying the priority of products or product groups and declaring them as urgent.

3. It supports rescheduling once a product is completed or a product/product group is inserted, removed or set as urgent.

4. The planning problems that emerge as a result of the modifications in (1)-(3) are all transformed into the Travelling Salesman Problem (TSP). TSP is a well-known and extensively investigated combinatorial optimization problem. By transforming various planning problems of chemicals manufacturers into a single structure, we are able to solve them all through only solving a new TSP instance that approximates the original sequencing problem, and avoiding to implement different algorithms for each problem.

5. The software effectively visualizes any given production schedule (product sequence) by indicating groups, differentiating fixed and flexible sequences within groups, highlighting required washes (mixer setups) and interactively displaying product attributes.

In this study, the combination of all these five features is presented within an integrated framework and a decision support system applied in the real world.

In the presence of current global efforts for efficient use of water resources, we believe that our work can also be applied to other companies operating in the chemicals industry and similar industries. Therefore, we have decided to openly share the software, its complete source code and documentation in multiple languages through the Internet. We hope that large scale environmental and economic benefits will be achieved through worldwide adoption of the software.

2.2 Literature Survey

Finding the product sequence with the least water consumption at a mixer is in fact the famous combinatorial optimization problem of *Travelling Salesman Problem* (TSP). In TSP, each node of a given graph has to be visited exactly once on a single tour, with the objective of minimizing the tour length. In our sequencing problem, products in the daily or weekly production list must be sequenced in such a way that the cost of changeovers is minimized. In other words, the total amount of water consumed for washing the mixer has to be minimized. A detailed analysis of TSP, including historical notes, various formulations and solution algorithms can be found in many classic operations research books: Aarts and Lenstra (1997), Bramel and Simchi-Levi (1997), Korte and Vygen (2005), Gutin and Punnen (2002), Toth and Vigo (2001).

TSP is typically a subproblem in production sequencing/scheduling problems in manufacturing facilities. There is a vast literature related to our study, thus we will mention only a small sample of these: Similar to the problem addressed in our paper, Pizzolato and Canen (2001) discuss how they improve the industrial competitiveness of the Brazilian subsidiary of a multinational chemical company. They develop a TSP-based model, focusing on the minimization of setup times and tardiness penalties through effective production sequencing. Sundararajan et al. (1998) describe the development and deployment of an optimization-based DSS for scheduling at a food processing company. Berlin et al. (2007) design a method to calculate a sequence of a given set of yoghurt products in order to minimize milk waste. They describe the construction of a practical method to achieve the environmentally optimal or close to optimal sequence of products through an optimal or heuristic algorithm that is selected based on the number of products to be sequenced.

Parallel to the growing public attention to the environmental problems and natural resource conservation, it is possible to observe an increase in the number of studies combining environmental perspective with manufacturing processes in the last decade. Effective scheduling of a large-scale paint production system is discussed by Adonyi et al. (2008). They extended a previously developed combinatorial optimization framework to generate a solution with minimal cleaning cost, which is the main source of waste generation in paint production.

Management of water use in chemicals industries is also discussed in literature. Wang and Smith (1994) address the minimization of wasted water in the process industries. They devise two design procedures where the first one maximizes driving forces in individual processes and the second minimizes the number of water sources for each process. While the existing methodologies for minimizing waste water are mostly for continuous processes, Almató et al. (1997) suggest a methodology which identifies the in-plant water reuse opportunities, resulting in significant reductions in fresh water consumption in discontinous, batch processes. They illustrate the methodology with a case study from food industry. Grau et al. (1995) discuss the product changeover waste which is processing-sequence dependent. They propose a procedure to reduce the amount of this waste, based on the generation of non permutation schedules in which the processing order of products may be different in all the recipe stages, and passing of products between stages is permitted. Almató et al. (1999) work on modelling, simulation and optimization of a water reuse system and they develop a software tool to implement the methodology. Despite our extensive literature search, we have not been able to find a work in the literature that exhibits more than two of the five distinctive characteristics that our system possesses.

Integration of manufacturing practices with environmental decision support systems significantly affects the state-of-the-art production techniques. Denzer (2005) discusses what Environmental Information Systems (EIS) and Environmental Decision Support Systems (EDSS) are and how these tools can be combined in a generic way. Matthies et al. (2007) provide the background of latest developments in EDSS. They state that better incorporation of interdisciplinary data, user-friendliness and integraton of visualization tools appear as general trend in EDSS development. Rivas et al. (2008) gives the mathematical basis and some examples of a model-based decision tool for the calculation of optimum design parameters in modern Wastewater Treatment Plants (WWTP). They formulate the optimum dimensioning of the biological treatment in WWTP as a mathematical optimization problem. Abou Najm et al. (1997) provide insights of an optimization model for integrated solid waste management. They present a user friendly computer-based interface implemented in MS Excel - Visual Basic environment.

Considering the environmental benefits resulting from the integration of information systems and operations research techniques, we believe that industrial applications contributing to the protection of environment and natural resources deserve more attention, especially in forthcoming years where the effects of global warming will be felt at much higher levels.

2.3 The Problem

2.3.1 The Underlying Travelling Salesman Problem

We model the production sequencing problem as a Travelling Salesman Problem (TSP) as follows: Each product in the production list is considered as a node in the TSP graph. The objective is to find the sequence of the products of the production list (the single tour that visits all the nodes in the graph) with the minimum water consumption (with the smallest travel distance). Given the chemical characteristics of each product, we compute the costs of changeovers between each pair of products based on how they interact chemically. These costs are considered as costs of traversing the arcs in the TSP graph. We reflect the grouping of products and the re-positioning due to urgency through modifying the representative graph, as demonstrated later in this section.

Our system handles fourteen different cases (Table 2.1) which arise due to having fixed, flexible or free products, and due to the presence or absence of urgency. In Table 2.1, FI, FL and FR denote the existence of fixed groups, flexible groups and free products in the production list, respectively. UR and NU denote the presence or absence of urgency, respectively.

A fixed group is represented with two dummy nodes: The first product of the fixed group and the last product of this fixed group. That is, only these two nodes are used to connect the group with the remaining of the model and the group's

Table 2.1: Characteristics of cases handled by DSS

	NU	UR
FI	Case 1	Case 8
FL	Case 2	Case 9
FR	Case 3	Case 10
FI, FL	Case 4	Case 11
FI, FR	Case 5	Case 12
FL, FR	Case 6	Case 13
FI, FL, FR	Case 7	Case 14

order is kept unchanged in the final sequence. In order to represent a flexible group, a sink and a source node are introduced to connect the group with the remaining of the model. The introduction of an urgent product or group is implemented by simply adjusting the costs associated with that item in such a way that it is forced to be processed at the beginning of the sequence. As a result, the number of nodes of the final TSP graph is a function of the number of free products, the number of fixed and flexible groups, the presence or absence of urgency and the condition of the mixer at the instant of production sequencing.

Let's consider a sample production list which includes a free products, one fixed group and one flexible group. This corresponds to a special instance of Case 7 in (Table 2.1). Let the number of products within the fixed and flexible groups be b and c, respectively. $G(N, A)$ represents the network over which the TSP model is defined. N and A are the sets of nodes and arcs and n is the total number of products:

$$n = a + b + c \tag{2.1}$$

Without loss of generality, the nodes in $G(N, A)$ are relabeled as given in Table 2.2.

Then we follow the algorithm in Table 2.3 to find the solution of the product sequencing problem. In Step (A3) of the solution algorithm SOLVE() of Table 2.3 we define the costs in the augmented network.

Table 2.2: Relabelling the Nodes

RELABEL() /* Relabelling the Nodes */

Dummy Origin/Destination (O/D) node: 0

Last product completed at the mixer before the sequencing decision is to be made: $0'$

Free nodes:
$NA = \{i : i = 1, \ldots, a\} = \{1, \ldots, a\}$

Nodes in the fixed group, sorted according to their given sequence within the group:
$NB = \{a + j : j = 1, \ldots, b\} = \{a + 1, \ldots, a + b\}$

Nodes in the flexible group:

$$NC = \{a + b + k : k = 1, \ldots, c\} = \{a + b + 1, \ldots, a + b + c\}$$
$$= \{a + b + 1, \ldots, n\}$$

Dummy fixed group node: $n + 1$

Dummy source node for flexible group: $n + 2$

Dummy sink node for flexible group: $n + 3$

$u_{i,j}$: Distance in the original TSP network from node i to node j (amount of water consumed for changeover from product i to j)

$w_{i,j}$: Distance in the augmented TSP network from node i to node j (amount of water consumed for changeover from product i to j)

V_j: Amount of water consumed for changeover from product $0'$ to j

Table 2.3: Solving the Product Sequencing Problem

SOLVE() /* Solving the Product Sequencing Problem */

(A0) Add new dummy nodes to the original network to obtain the augmented network:

$$N_1 = N \cup \{0, 0', n+1, n+2, n+3\}$$

(A1) Relabel the nodes in the augmented network according to the scheme RELABEL() described above.

(A2) Remove the nodes in the fixed group and all the arcs linked to them from the augmented network

$$N_2 = N_1 - NB$$

(A3) Define the new costs according to the scheme below to obtain the final network (N_3).

(A4) Solve the TSP on the final network N_3.

Table 2.4 describes the algorithm that performs the modification of costs. The integration of flexible groups within the model assumes an approximation of cost coefficients.

Table 2.4: Defining the new costs

DEFINE NEW COSTS() /* Defining the new costs */

(B1) $w_{i,j} = u_{i,j}, \forall (i,j) \in N_2 \times N_2$

Continued on next page...

Table 2.4 Continued

(B2) It costs nothing to enter into or exit from the dummy O/D node 0: $w_{0,j} = w_{j,0} = 0, \forall j \in N2$

(B3)

a. Production cannot start with the dummy O/D node:

$w_{0',0} = \infty$

b. Cost of starting the sequence with product j:

$w_{0',j} = V_j, \forall j \in N_2 - 0$ (This cost is due to the wash when changing from $0'$ to j.)

c. No node except the dummy O/D node can precede $0'$:

$w_{j,0'} = \infty, \forall j \in N_2 - 0$

(B4) Costs for the fixed group:

a. The cost of entering the dummy fixed group node $n+1$ is equal to the cost of moving to node $a+1$ (the first node in the fixed group) of the original network:

$w_{i,n+1} = u_{i,a+1}, \forall i \in N_2$

b. The cost of leaving the dummy fixed group node $n+1$ is equal to the cost of moving to node $a+b$ (the last node in the fixed group) of the original network plus the cumulative cost incurred within this fixed group:

$w_{n+1,j} = u_{a+b,j} + \sum_{i=a+1}^{a+b-1} u_{i,i+1}, \forall i, j \in N_2$

Continued on next page...

Table 2.4 Continued

(B5) Costs for the flexible group:

a. *i.* A node within the flexible group cannot be preceded by nodes outside the flexible group (NC):

$w_{i,j} = \infty, i \in N_2 - NC, j \in NC$

ii. A node within the flexible group cannot precede by nodes outside the flexible group (NC):

$w_{j,i} = \infty, i \in N_2 - NC, j \in NC$

b. *i.* It costs nothing to enter the source node $n+2$ of the flexible group (NC):

$w_{i,n+2} = 0, \forall i \in N_2 - NC$

ii. The source node $n + 2$ of the flexible group (NC) cannot precede an outside node:

$w_{n+2,i} = \infty, \forall i \in N_2 - NC$

c. *i.* It costs nothing to exit from the sink node $n + 3$ of the flexible group (NC):

$w_{n+3,j} = 0, \forall j \in N_2 - NC$

ii. A node outside the flexible group cannot precede the sink node $n + 3$:

$w_{j,n+3} = \infty, \forall j \in N_2 - NC$

Continued on next page...

Table 2.4 Continued

d. A node within the flexible group cannot precede the source node $n+2$ or succeed the sink node $n+3$:

$w_{i,n+2} = \infty, \forall i \in NC$

$w_{n+3,i} = \infty, \forall i \in NC$

e. *i.* It costs nothing to move from the source node $n+2$ to a node within the flexible group (NC):

$w_{n+2,i} = 0, \forall i \in NC$

ii. It costs nothing to move from a node within the flexible group (NC) to the sink node $n+3$:

$w_{i,n+3} = 0, \forall i \in NC$

As noticed in Table 2.4, the changeover cost between a product outside a flexible group and a product inside the group is ignored. The flexible groups are handled by the addition of dummy source and dummy sink nodes for each flexible group. A sequence of nodes including a flexible group is formed by entering to the flexible group through the dummy source node and exiting the group through the dummy sink node only. In other words, once a flexible group node is added to the sequence, all other nodes within this group must be visited before exiting the group and the changeover costs for flexible groups relies on this assumption.

2.3.2 The Solution Approach

Two different solution methods are implemented in our system depending on the number of nodes in the TSP model. For "large" number of nodes, a "good" solution is obtained by simulated annealing, while for "small" number of nodes a near-optimal solution is found in reasonable computation time considering the approximate flexible group costs. The decision support system selects the appropriate method by checking the number of products and the grouping structures specified by the production planner.

Solving Small Instances

Once the TSP model is constructed, if the number of nodes in the model is less than 15, then the optimal production sequence is computed for the MIP model with approximate cost structure for the flexible groups. Our decision support system constructs the Mixed Integer Programming (MIP) model for a TSP instance by calling the open source lp_solve library (lp_solve). The system uses the formulation in page 529 of Winston (1994), which is presented in the Appendix. The optimal solution for small TSP instances (instances with less than 15 nodes) can be solved in at most 2 seconds.

Solving Large Instances

Since it takes extremely long run times to solve the TSP instances with large number of nodes to optimality, the alternative approach of using a *simulated annealing* metaheuristic is preferred for such instances. Simulated Annealing (SA) is an artificial intelligence based algorithm which searches for a better solution at each iteration while still allowing a transition to an inferior solution with a certain probability. Even though the algorithm always accepts an improving solution, it can also accept the move to an inferior solution with a probability that is derived from Maxwell-Boltzman distribution given in 2.2.

$$p\left(\delta E\right) = \exp\left(\frac{-\delta E}{T}\right) \tag{2.2}$$

Here δE refers to the last improvement in the objective function while T refers to the temperature at the current iteration. At higher temperatures the p value is higher and a large region in the solution space can be traversed. This enables diversification and thus avoidance of local minima. As the temperature is gradually decreased in each successive iteration of the SA algorithm, p eventually becomes much lower. Hence, the algorithm gives priority to improving near a local minimum.

Since simulated annealing is a probabilistic algorithm, it may come up with a different sequence each time the program is run. Neighbourhood structure in this technique is such that two random numbers corresponding to two nodes are generated and these two nodes are swapped pair wise to obtain a candidate neighbour. We have used a publicly available simulated annealing code, which is implemented by Chalhoub (2006) in C# in our system after slight modifications. In our case, the SA algorithm was applied to the production planning practice of JohnsonDiversey Turkey, and was able to reach "very good" solutions[1] in negligible computation time compared to the previous practices.

2.4 The Framework

The UML (Unified Modelling Language) activity diagram given in Figure 2.1 depicts the flow of actions taken by the DSS and the user in a structured framework. According to Schmuller (2001), "the UML is a visual modeling language that enables system builders to create blueprints that capture their visions in a standard, easy-tounderstand way, and provides a mechanism to effectively share and communicate these visions with others." We believe that far from being just a visual tool, the UML diagrams are extremely useful for model implementation and documentation. In order to do so, we use UML diagrams in our study to clearly express the model independent from a programming language, helping the users easily understand and interpret the software framework.

In the literature, the UML diagrams are used in a wide range from user interface and process modelling to production planning and control. Papagjorji and Shatar (2004) discuss the UML as a methodology to simplify documentation of model requirements, assumptions and calculations as well as a template for implementing the model in programming languages. Tsai and Sato (2004) provide a UML model of an agile production planning and control system (APPCS) which is a system for planning, scheduling, procurement and production control. Muzy et al. (2005) use the UML in the modelling and simulation of fire spread, as a common way of communication between programming specialists and domain experts.

Activity diagrams are about the activities that cause a change in the object

[1]As quoted from the production planners at the company

states. OMG (Object Management Group) (2003) defines an activity graph as "a variation of a state machine in which the states represent the performance of actions or subactivities and the transitions are triggered by the completion of the actions or subactivities." The objective of this diagram is to convey the information related with process flows while considering the actions undertaken by each component in the system (i.e. the user and the DSS in our model). In Figure 2.1, the diagram is seperated into parallel segments called *swimlanes*. Thus, an additional dimension is added to the model to visualize the roles in the model. The UML activity diagram given in Figure 2.1 is designed to summarize what happens after the software is initialized by the user.

Transitions from one activity to another are shown with solid arrows which reflect like logical paths followed during the use of software. For instance, after the user starts the software, the DSS reads the database to display the products stored in the current database on the main window. Then, it waits until an action is performed by the user. Once the user decides to continue using the software, any product(s) in the production list is selected from the current database by the user. At this moment, the DSS gives a decision about the status of the current production list. This decision is illustrated with a small diamond which has the possible paths flowing out of the diamond.

Once the status (i.e. whether the list is empty or not) of the current production list is evaluated by the DSS, the user is free to choose different actions depending on the decision in the previous step. To be specific, while the production list is empty, the user can also add product or product groups to the production list. On the other hand, if the production list is not empty, the user is free to :

1. add a selected product into the current production list
2. add selected products into the production list as a group
3. solve for a new product sequence
4. display the properties of a selected product from the current production list
5. delete a selected product from the production list
6. visualize the current production list

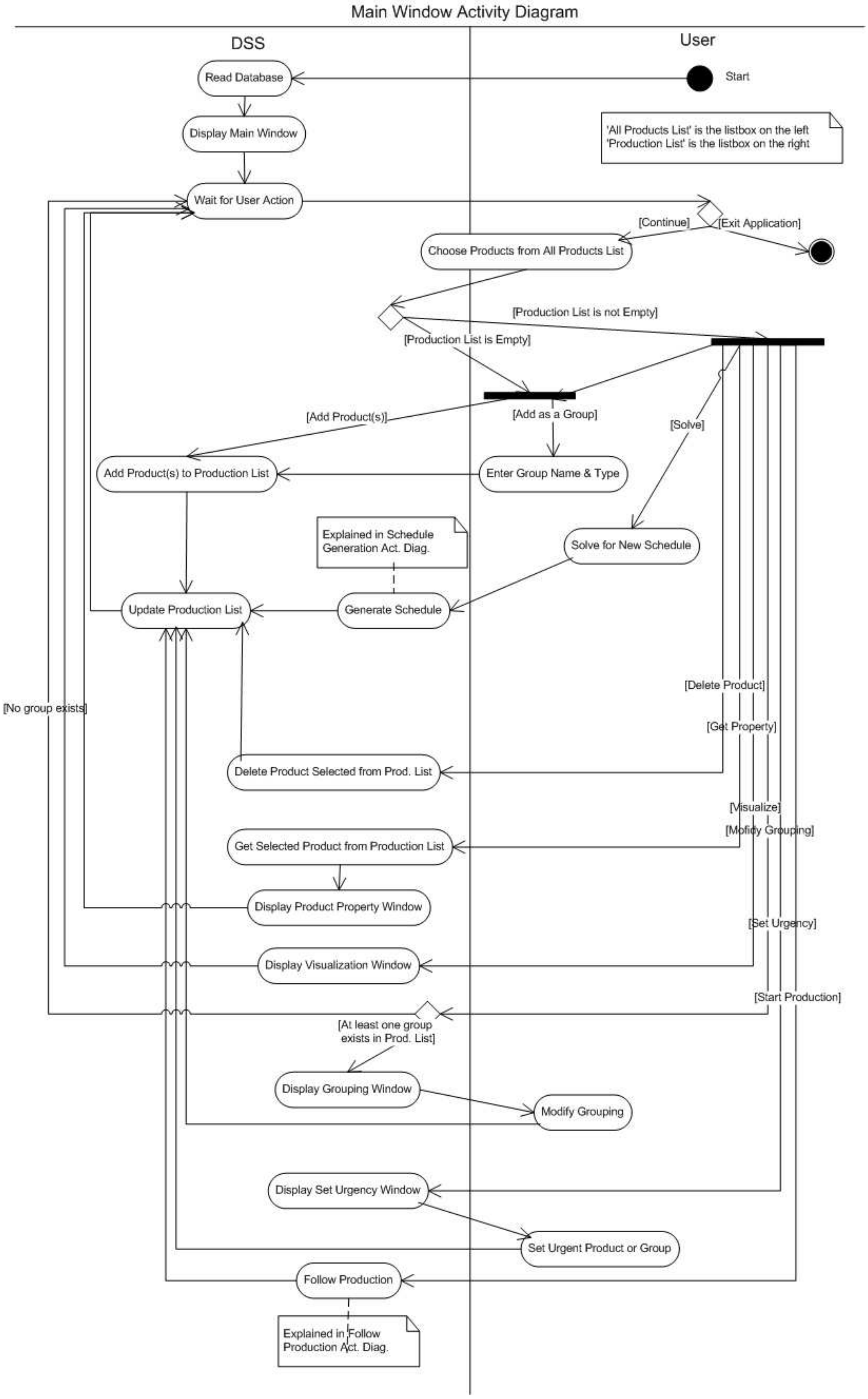

Figure 2.1: UML activity diagram for the main window

7. modify the product groups if available

8. set a product or product group as the urgent, and

9. start and track the production.

Any of these actions can be performed as the next action where the current production list is not empty. Thus, these actions are represented with concurrent paths in the UML diagram. Figure 2.1 gives a roadmap of the subsequent activities to be performed when any of the paths mentioned above is followed. For instance, if the user selects to add products as a group, the next action to be performed is entering the group name and the group type (i.e. flexible or fixed order). For further details about the activities performed in the main window of the software framework, please refer to Figure 2.1.

In Figure 2.2, we provide the UML activity diagram of the new schedule generation within the sofware framework. After the user clicks to solve for a new schedule, the DSS immediately asks for the status of the mixer from the user. The user provides the information whether the production has already started (i.e. there are some residues in the mixer from the previous products). How the DSS handles the schedule generation process is visually explained in detail in the remaining parts of the UML activity diagram in Figure 2.2.

After the number of product groups, properties of these groups and the number of the free products are taken into account concurrently, the modified array structure representing the current production list is formed. This modified array structure can be thought as a host for the final TSP model. In order to fill this modified array structure, the first thing the DSS performs is simply creating two counters i and j, standing for the free products and product groups, respectively. For each different product, in other words for each value of i, the UML activity diagram in Figure 2.2 shows how each product or product group is represented in the final TSP model.

Once the TSP model and associated cost structure are completely built, counter i exceeeds the number of products in the system, enforcing the software framework to check for the number of nodes in the final TSP model. If this number is larger than 15, a heuristic technique is used while the problem size is smaller than 15 the problem is solved optimally by calling lp_solve dynamic link library file. The threshold level for solution technique, 15, is a previously set value and can be adjusted in accordance with production characteristics and the computational capability that is available in the facility. It's important to note that the optimal solution to the TSP is not the optimal solution to the sequencing problem, since the TSP graph is an approximation of the original graph. In the approximate TSP graph, the costs into and out of the flexible products are ignored.

Figure 2.3 enables us to visually represent the flow of tracking and control activities during the execution of the production process. Production tracking is a manual process controlled by the mixer operator. After the new product sequence is generated, the mixer operator starts the production in the mixer. Each time a product is started and completed, the times of these events are recorded by means of the DSS and user interaction. This interaction is provided with a button located in the production tracking window of the DSS. Furthermore, the operator can interrupt the production process in case of a need for resequencing of the products such as a new order arrival, change in urgent product status and product group modification. The UML activity diagram in Figure 2.3 summarizes the production tracking process.

2.5 The Decision Support System

The coding of the decision support system, which is a stand-alone Windows application, is completed in the C# language under MS Visual Studio.NET. The selection of the programming language and the development environment was made in accordance with the established information technology (IT) infrastructure of JohnsonDiversey Turkey, the company for which we apply our general software framework. When the software is initialized, it automatically reads the product information from a MS Access database file.

For a particular product, the *PRODUCT* table in the database stores the values for the fields *ProductID*, *ProductName*, *ProductNo* and *SequenceID*. *ProductID* is the key for the *PRODUCT* table and the *SequenceID* field is linked to data fields in another table that stores chemical properties. When needed, the software interacts with the database to keep the log of production records and to modify product characteristics. The user can add or delete products through the graphical user

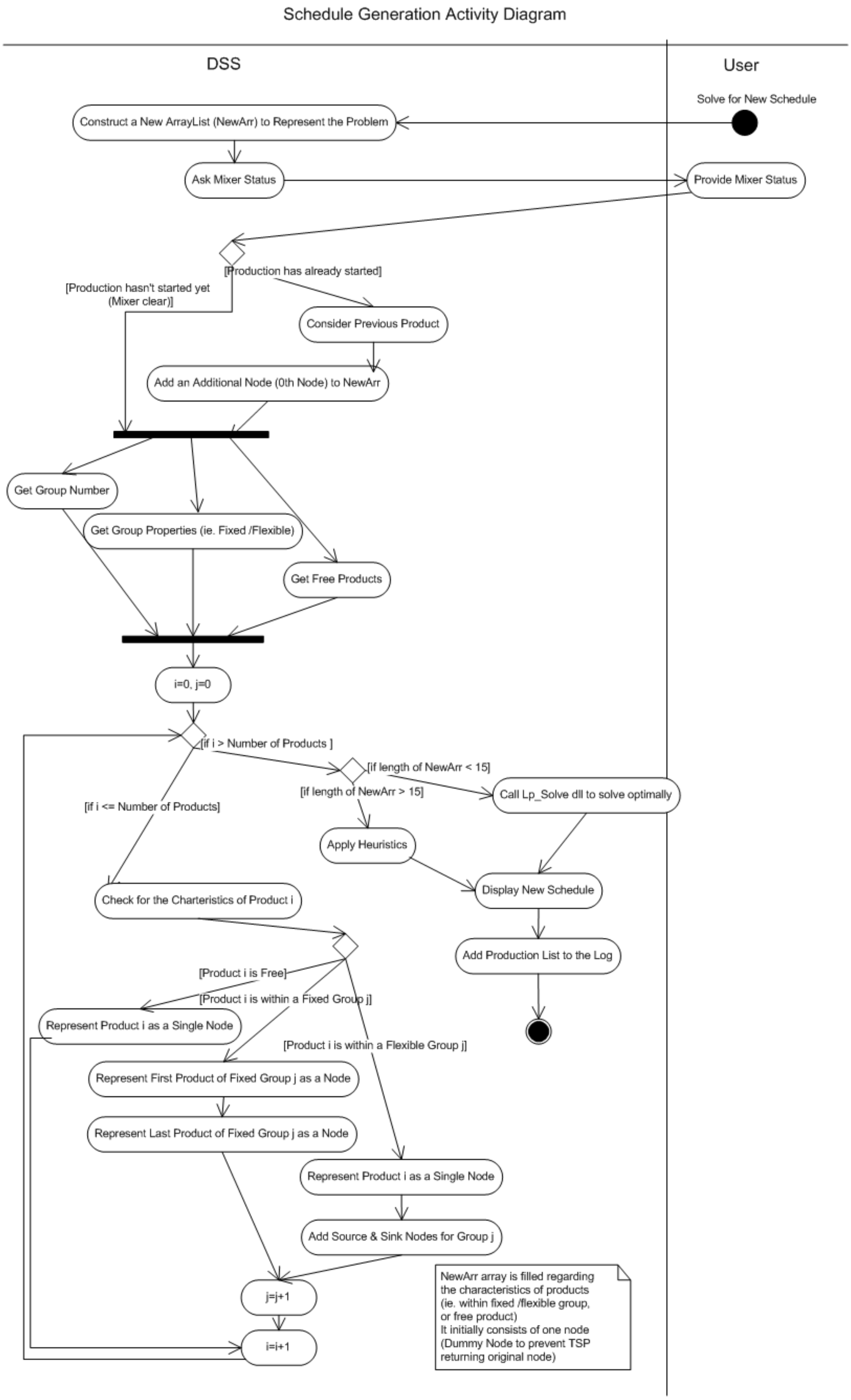

Figure 2.2: UML activity diagram for schedule generation

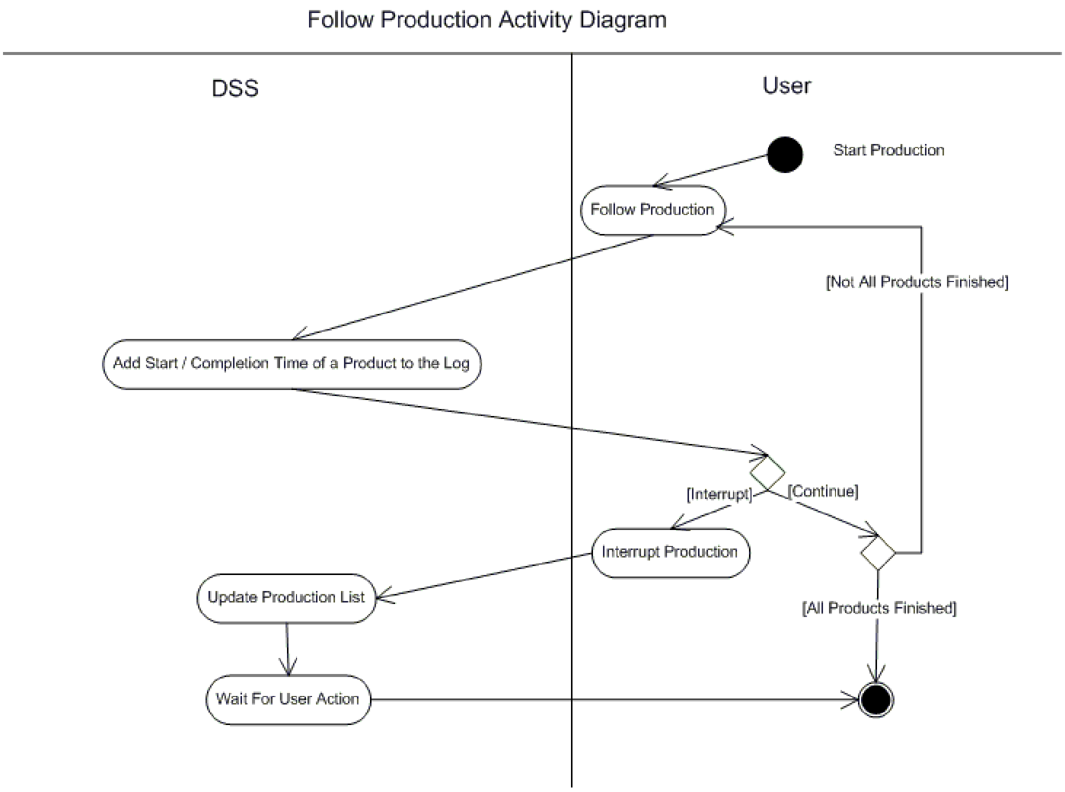

Figure 2.3: UML activity diagram for tracking production

interface (GUI) of the DSS.

2.6 A Case Study: JohnsonDiversey Turkey

In this section, we describe the implementation of the decision support system (DSS) for saving water by effective product sequencing at JohnsonDiversey Turkey (JohnsonDiversey), the local branch of the global cleaning and hygiene products manufacturer, JohnsonDiversey. Located in the industrial city of Kocaeli, the production plant manufactures liquid and powder products. Liquid products are produced in large mixers, varying in type and sizes, by mixing pre-specified quantities of the constituent ingredients for specific time periods. Each product family is allocated to a particular mixer based on its chemical characteristics such as containing phosphorous/silicate or having foaming property. The number of products for a typical sequencing problem encountered in the weekly production planning of the company generally does not exceed 10. However, in the peak periods (i.e. summer months due to large volume of demand from hotels), this number can increase substantially.

While producing two different products at the same mixer, it may be required to wash the mixer with clean water during product changeover to prevent any contamination effects. The weekly schedules (product sequences) for the mixers have traditionally been determined by the planners in the facility, based on their experience and wisdom. Early in our study, we have noticed that a smart method for sequencing the products at the mixers could result in significant savings in mixer setup times, labor requirements and water consumption. Our study eventually evolved into the current system, namely “SmartScheduler”, that we describe in the paper.

Lastly, it is important to bear in mind that water is becoming an increasingly scarce and strategic resource in Turkey, as well as the world. Meanwhile, companies operating in Turkey feel the pressure to conduct their operations in more environmentally friendly ways compared to the past. This transition is motivated by the urge to conform with the European Union (EU) legislations, as a part of Turkey’s candidacy process for EU. The described DSS can aid the local chemical production facilities easen to comply with such environmental regulations.

2.6.1 Production Planning at JohnsonDiversey Turkey

JohnsonDiversey Turkey has experienced a dramatic increase in demand within the last two years and the manager of the Kocaeli plant recognized the need for a more systematic approach for production planning. This need is especially stressed during periods of high seasonal demand in the domestic market.

Every Monday at 9 a.m., the production team at the plant assembles to discuss the weekly production requirements and determine the product sequence at each mixer for the week. Currently, the schedules of all the mixers -except the mixer considered in our study- are determined through rules of thumb. If an urgent order is received by the company during the week, then the previously-established weekly schedule is interrupted and the urgent products are processed as soon as the mixer is released of the already-started product. The production planners also prefer to be able to group particular products in fixed or flexible groups. This gives the planners the freedom to postpone the products with large processing times, and to bring front the ones that are urgent.

Each product has its own characteristics in terms of odor, color, foaming and existence of critical chemical substances such as phosphorus or silicate. In our database a specific numeric label uniquely identifies each product based on these characteristics. Such characteristics determine whether a mixer wash will be required between processing of two successive products. A typical wash cycle takes approximately five minutes and requires 360 litres of clean water.

At the time of writing of this paper, our decision support system is being executed for a single mixer and is already enabling significant savings for the selected mixer. Even though the next step of our project was planned to be applying the system to all mixers in the plant, the project was discontinued since the company started using a more extensive commercial software in planning its operations starting in March 2008.

2.6.2 Implementation of the DSS

In order to solve the product sequencing problem of JohnsonDiversey Turkey, we make our general product sequencing framework applicable to the production facility of JohnsonDiversey Turkey. The decision support system based on this sequencing framework is built as an executable application, and it is started from the desktop of a personal computer located in the production planning department. The software is currently deployed on the computers of the production planners at the Kocaeli plant of JohnsonDiversey Turkey. The software has been actively used in production planning approximately seven months beginning from September 2007 to March 2008.

The complete set of products assigned to the mixer is displayed inside the list box on the left hand side of the main window (Figure 2.4). The planner forms the production list by selecting products to be produced in a week from this list box. The production list is displayed inside the list box on the right hand side of the main window.

Free products which are not included in a product group are added to the production list by clicking to the *Add Product* button. Products that belong to a group are added to the list by first clicking the Add as a Group button and then providing

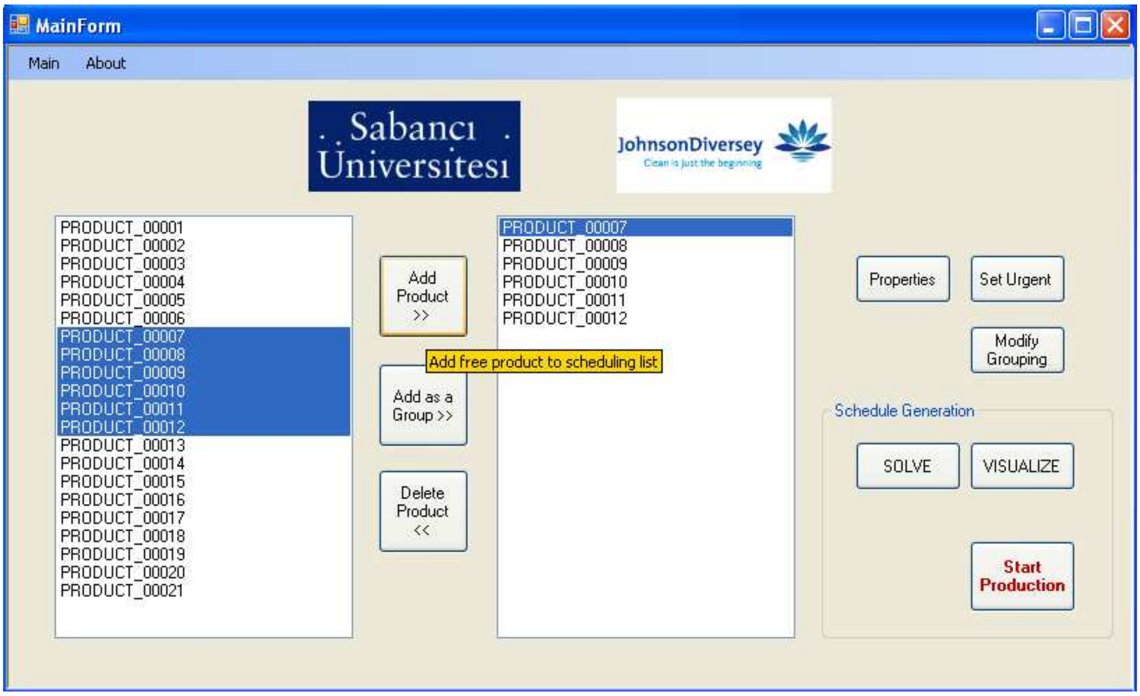

Figure 2.4: Adding a free product to the production list

the relevant group information (Figure 2.5). A product group can be set as fixed or flexible, denoting whether products inside the group should be fixed or not. The user can insert additional information about the group into the optional "explanation" area in the grouping window. Furthermore, a large number of characteristics belonging to product group can be changed by the user using the *Modify Grouping* window of the software.

Figure 2.6 shows an instance related to a sample product group, named "Customer A". The user can select the product group to be modified from the dropdown menu located at the left-top section of the window. All items within the selected group are displayed on the left listbox. In the fixed group "Customer A", it is clear that there are three different products (i.e. Product00006, Product00007 and Product00008). The user can adjust the order of these products using *Up* and *Down* buttons located at the left part of the window. In addition, the user can add a product to this group from the current database by selecting the product from the right listbox and clicking on the *Add* button. The user can remove a product from the selected group in a similar manner by clicking on *Delete* button.

The current status (i.e. fixed or flexible) of the selected group is shown at the

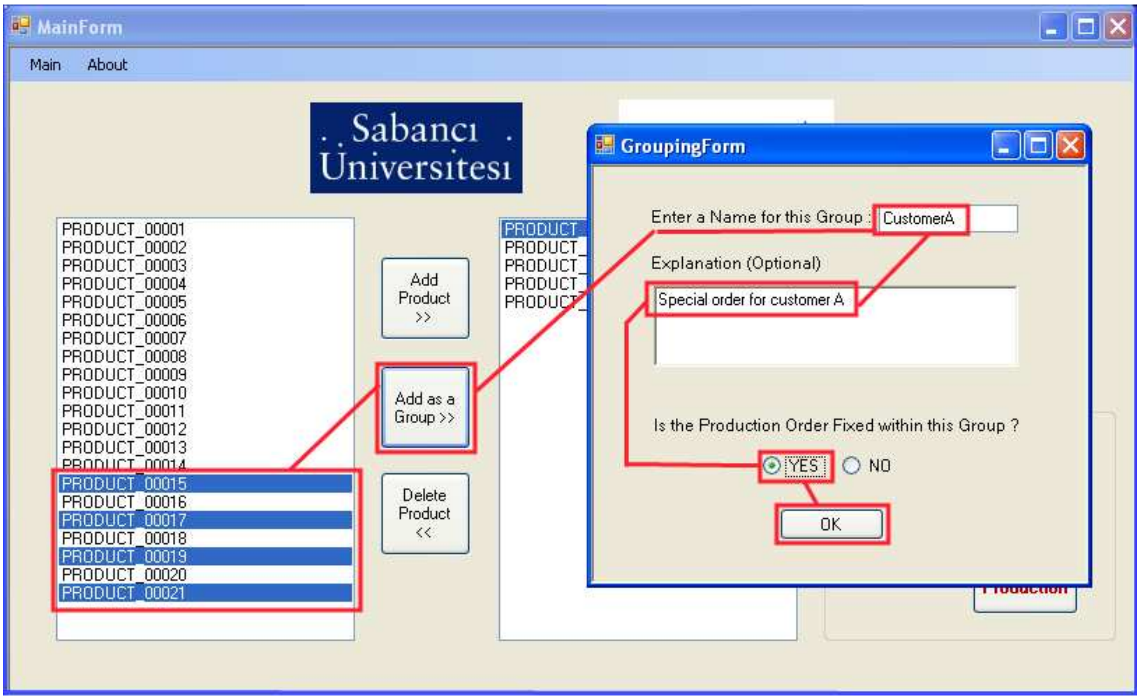

Figure 2.5: Adding a product group to the production list

end of the window together with a button which enables the user change the status of the group. The user is capable of disposing a product group. In this case, all items within the disposed group are labeled as free products.

The user can generate a solution (product sequence) in the DSS by clicking the *SOLVE* button (Figure 2.4). The initial state of the mixer (clean or with residue) affects the solution, so this state is questioned by the DSS. According to the number of nodes in the resulting TSP model, the appropriate solution method is selected and applied. The solution (beit, near optimal, or just "good") is generated in negligible computational time by the DSS and is displayed inside the production list box. By clicking the *VISUALIZE* button (Figure 2.4) the planner can visualize any sequence, including the solution generated by the DSS, in more detail in a new window (Figure 2.7).

The visualization window displays the products in the production list in their ordered sequence, together with their chemical characteristics. Planned washes are indicated by solid vertical red lines between successive products. Product groups are distinguished by shading, and a selected group is highlighted with yellow color. Figure 2.8 shows the initial product sequence for the problem in Figure 2.7 in the

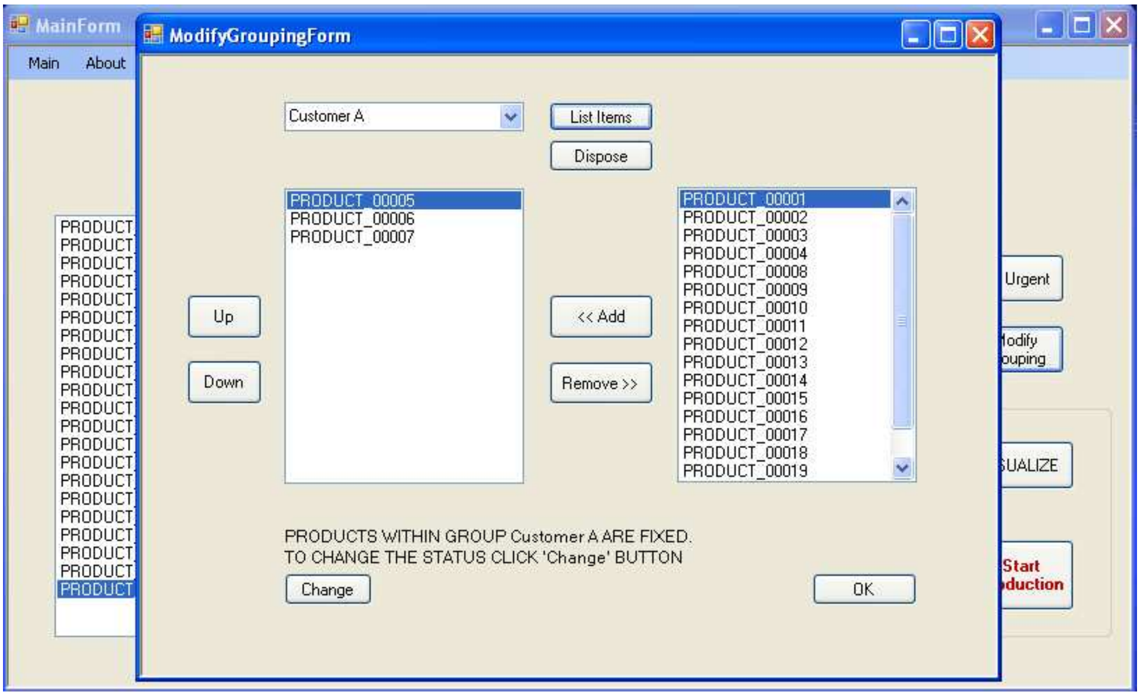

Figure 2.6: Modifying product groups

visualization window. The visualization enables immediate discovery of critical information, including the fact that the DSS has reduced the number of required washes from four to one, saving more than one ton of water in this example. The DSS illustrates the urgent order with red color in the visualization window.

The operator of the mixer can also track the production step by step after *Start Production* button is clicked. The DSS displays the status of the current production in the mixer on a separate production window (Figure 2.9). The operator clicks on *OK* button to record the daily activities on the production log (i.e. the start and finish time for a particular product). Once an urgent product is received, the operator interrupts the production, and the solution generation process is repeated with the new production list that includes the recently added urgent order.

Whenever a new product sequence is generated and the processing of a product is started or completed, the time of the activity and associated products are recorded in the production logs. A separate MS Access database file stores these production logs. The availability of such records makes it possible to investigate the production trends within a given time interval at the plant.

JohnsonDiversey Turkey used the decision support system effectively in produc-

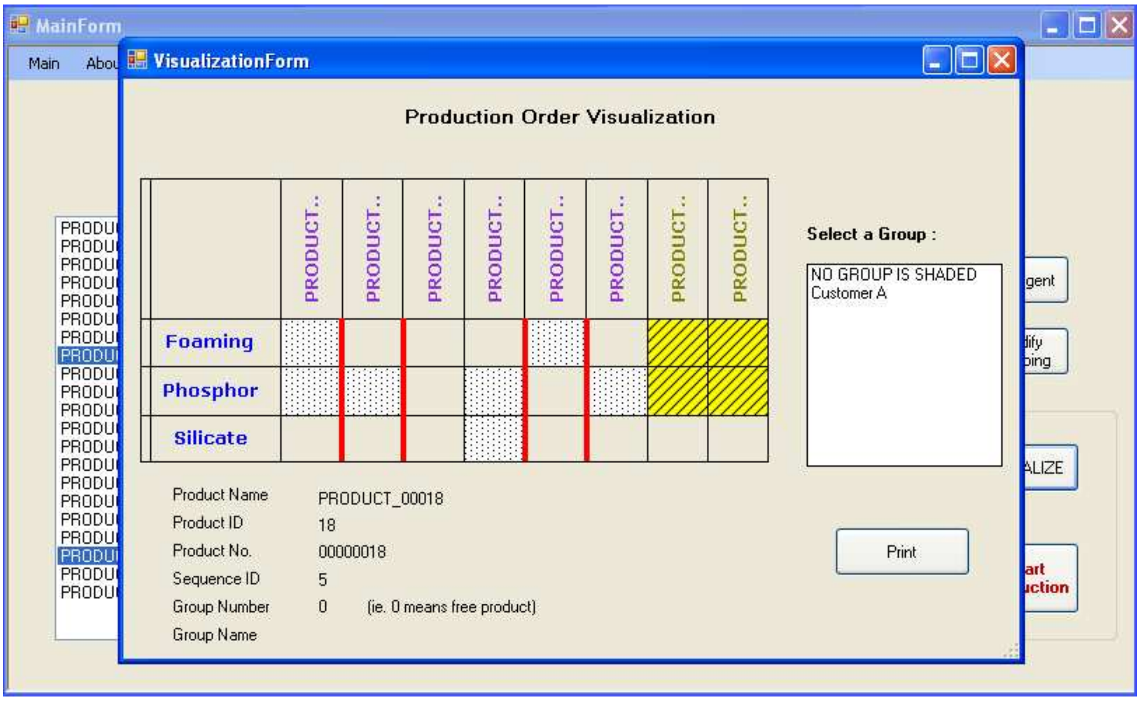

Figure 2.7: Visualization of the initial production sequence

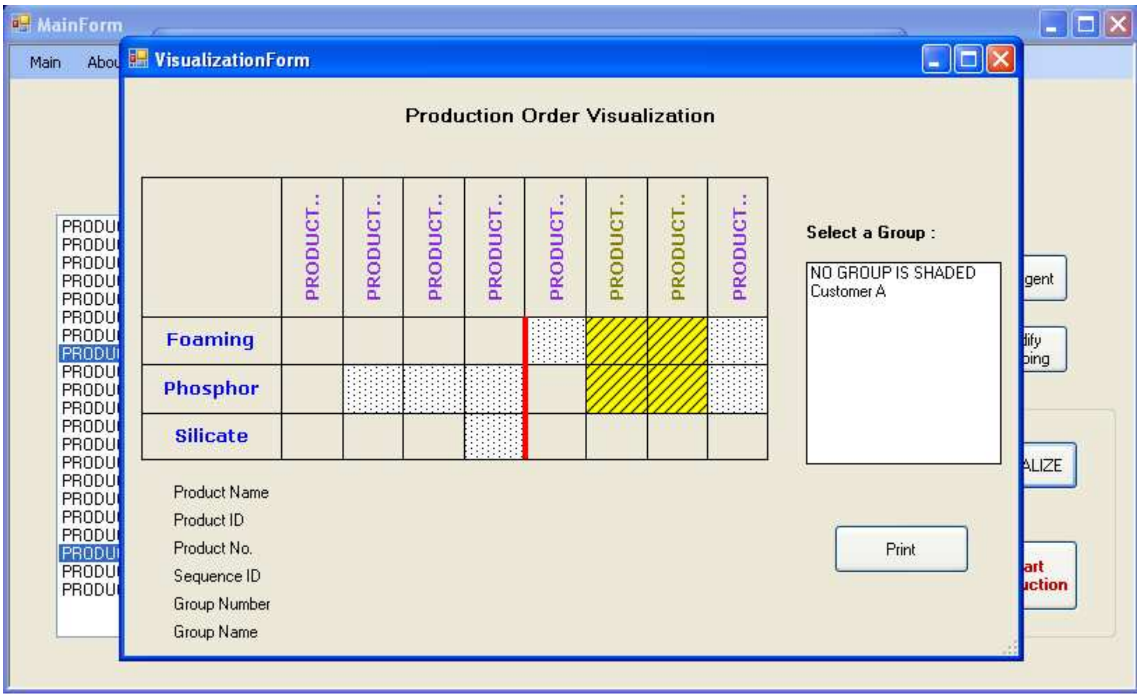

Figure 2.8: Visualization of the best sequences

Figure 2.9: Production window to track current mixer status

tion planning approximately seven months. Compared to the previous product sequencing practices which highly depend on the experience and wisdom of the mixer operator, significant amount of clean water is saved due to the reduced number of required washes during product changeovers. For instance, although the production was not in its peak season during the testing period of our software, approximately 700 liters of clean water is saved in the first few days of implementation. Considering the substantial amount of water savings in production, the production planners decided to regularly use the product sequencing system until the company switched to the other production planning software that is used globally throughout the JohnsonDiversey.

2.7 Conclusion and Future Work

Optimization-based decision support systems can bring significant benefits to manufacturing companies from an environmental point of view. In this paper, we have summarized our work towards saving water at a chemical products plant. The system that we have designed and developed has been regularly used by production planners at a chemical plant, serving as an example application of environmental operations research (OR) in Turkey. By effective use of the DSS, not only less clean water consumption but also savings in energy, waste water, time (i.e. more production can now be scheduled in the same amount of time) are expected to be achieved. We have released the source code of our software on the Internet with the aim of propagating the impact of our study and facilitating the application of similar systems throughout the world.

There are several avenues of future research that we plan to pursue:

- We plan to extend the application of our system to other companies in the industrial regions, which are home to hundreds of chemical manufacturers around Sabanci University.
- We plan to post extensive documentation in several languages at the project web site http://www.opendecisions.org. Currently, video tutorials in English, Turkish and Russian are available at the project web site.
- We plan to enable our software to handle the multi-mixer case, where products are allowed to be assigned to more than one mixer. This new setting corresponds to another well-known optimization problem, namely the Vehicle Routing Problem (VRP), which contains TSP as a subproblem.
- One other future project is to extend the system to handle various other types of issues such as due dates, resource constraints, uncertainty etc.
- Amount of cost savings can be quantified not only in clean water savings but also reductions in energy use, production time and waste water release.

Acknowledgements

The authors would like to thank Sabanci University alumni Melis Başaraner, Sedat Sever and Onur Sercan Şire for their help in the early phases of this project. The authors would also like to thank Altuğ Erbil, Arif Çepni and Mehmet Bozan at JohnsonDiversey Turkey for supporting the project and applying the developed decision support system in regular production scheduling at the Kocaeli production plant.

Appendix: Integer Programming Formulation of the TSP

Assume that the TSP consists of nodes $1, 2, 3, , N$. For ilet c_{ij} denote the distance from node i to node j and let $c_{ii} = M$, where M is a very large number (relative to the actual distances in the problem). Setting $c_{ii} = M$ assures that the tour will not include node i immediately after leaving node i. Meanwhile define

$$x_{ij} = \begin{cases} 1 & \text{if } i \text{ is succeeded by } j \text{ in the solution} \\ 0 & \text{otherwise} \end{cases}$$

Then the solution to a TSP can be found by solving

$\min z = \sum_i \sum_j c_{ij} x_{ij}$

s.t.

$\sum_{i=1}^{i=N} x_{ij} = 1, j = 1, 2, \ldots, N$

$\sum_{i=1}^{i=N} x_{ij} = 1, i = 1, 2, \ldots, N$

$u_i - u_j + N x_{ij} \leq N - 1, (i \neq j; i = 2, 3, \ldots, N; j = 2, 3, \ldots, N)$

x_{ij} binary, $u_j \geq 0, \forall i, j$

Chapter 3

A Software Framework for the Integration of Data Envelopment Analysis and Data Mining

In this study, the main objective is to develop a general framework to analyze the solutions of any Data Envelopment Analysis (DEA) model with a computer science and data mining point of view. The paper formally shows how the solutions of any DEA model should be structured so that these solutions can be examined and interpreted by analysts through information visualization and data mining techniques effectively. An innovative and convenient DEA solver, SmartDEA, is designed and developed in accordance with the proposed analysis framework. The developed software provides a DEA solution which is consistent with the framework and is ready-to-analyze with data mining tools, thanks to its table-based structure. The developed software and analysis framework are tested and applied in a real world project for benchmarking the dealers of a leading Turkish automotive company. For the confidentiality of the data, the results and analysis are not provided in this study, but the developed solver and the general framework of it are discussed in detail.

3.1 Introduction

Data Envelopment Analysis (DEA) is a widely used method in performance evaluation and benchmarking of a set of entities. Its convenience in assessing the multiple input and output variables of these entities by not requiring congruity and a priori relationship makes it a very popular management tool in many application areas. Another reason for its wide use is the managerial insights that come up with the solution of a DEA model. For instance, DEA assigns a peer group or reference set for an inefficient entity which can take the entities in this set as role models in accordance with the assigned weights. Another important DEA result is the target values or projections for the input and output variables of an inefficient entity to reach efficiency level. Similar to these outcomes of DEA, the method provides a significant amount of information from which analysts and managers derive insights and guidelines to promote their existing performances. Regarding to this fact, an effective and methodologic analysis and interpretation of DEA solutions are very critical.

In this study, the main contribution is to develop a general methodology to enable DEA analysts to extract the most important and interesting insights in a systematic manner. In order to do so, a computer science and data mining perspective is reflected on the designed format of the DEA results. Various data mining and information visualization techniques can be appropriate for the analysis of different types of DEA models. The paper provides a fundamental basis for the implementation of these techniques in the DEA solutions. A convenient and general notation is proposed for the DEA data included in the model, the other data and the results data generated by DEA solvers. The ultimate goal of the study is to build a structure framework for the analysis of DEA results, enabling researchers and practitioners to make analytical benchmarking and performance evaluations. For such an analytical tool, a structured representation of DEA results framework is needed.

In accordance with the proposed framework, a user-friendly and convenient DEA software, SmartDEA, is designed and developed. The software generates DEA solutions with a structure consistent with the framework. The stages fulfilled by a DEA analyst are simply importing the model data, solving it using the appropriate DEA model and making analytical inquires on the generated solution data to evaluate and benchmark the entities assessed in DEA model. The solution data generated by the software allows analysts to integrate the results and many of the data mining and information visualization techniques in a convenient and effective manner.

The main motivation in the development of the DEA software is a real world project for benchmarking the dealers of a leading Turkish automotive company. Using the real world data, dealers are evaluated comparatively and in automotive company's perspectives, respectively. In order to keep the name of the company confidential and for convenience, the ABC company will be used from now on to refer the automotive company.

In the testing of the software, the relationships of the ABC's spare parts warehouse and its dealers are analyzed in terms of some inputs and outputs such as total spare parts area of dealers, total dealer expenses, spare parts employees of dealers, total revenue of dealer and amount of purchase from the automotive company by the dealers. Two different DEA models are built and solved by using the SmartDEA for benchmarking the dealers of ABC to help managers and decision makers to set guidelines and take actions to enhance the overall business practice of the company. Since the data of the study is confidential, the developed DEA solver and its framework are mentioned in more detail than the results of the analysis.

3.2 Literature on Data Envelopment Analysis

In this section, the DEA literature will be discussed briefly to make the reader familiar with the methodology. Considering the wide range of applications that are addressed by DEA, only some of the studies related with the test model (i.e. comparing the efficiencies of dealers) of the SmartDEA and software framework are discussed. Gattoufi, Oral and Reisman (2004) presents a scheme for the classification of the DEA literature. The paper suggests that the articles can be investigated on the following criteria: (1) Data source, (2) Type of the implemented envelopment, (3) Analysis, and (4) Nature of the paper. For the content analysis of DEA literature and its comparison with operations research and management science fields, the readers can refer to the paper by Gattoufi et al. (2004).

DEA has been widely used as a strong performance evaluation tool in the literature. Weber (1996) shows how the application of the DEA can lead to significant fiscal savings as well as other measurable performance evaluation criteria. In the study, a DEA model is formulated to measure vendor efficiencies and the implementation of the technique in a baby food manufacturer is provided as a case study. Weber et al. (2000) integrates multi-objective programming and DEA to evaluate the number of vendors to employ. The authors initially solve for the number of vendors and then evaluates the efficiencies of them using multiple criteria. In the study, a case study is provided for a Fortune 500 company in a just-in-time (JIT) manufacturing environment. Another widely encountered problem type, supplier selection problem, is adressed by Liu et al. (2000). The authors presents an application of DEA to assess the overall performances of suppliers for a manufacturing company. Using a simplified model of DEA, the general performance of the suppliers are benchmarked with the strategic aim of reducing the number of suppliers effectively and providing the suppliers targets to improve their positions.

Easton et al. (2002) applied the DEA in purchasing management. The study performs a DEA to help managers improve the efficiency of the purchasing operations. The comparison of purchasing efficiencies of different companies operating in the petroleum industry is presented using a DEA model. In this study, a detailed management implication of DEA is given together with the method's strengths and limitations. Ross et al. (2002) provides an integrated benchmarking approach to distribution center performances using DEA modeling. Approximately 100 distribution centers are benchmarked and their productivities are examined by using extensive tools of DEA such as facet analysis and window analysis. After the role model distribution centers are determined, strategic managerial insights are obtained. The authors also discussed the strengths of DEA in environments where explicit knowledge about the relationship between the inputs and outputs is obscure. Talluri et al. (2006) evaluates the performance of vendors using DEA but with a significant variation in the model. In order to get a precise assessment of vendor performances, the authors attempt to consider variability in vendor attributes by suggesting a chance-constrained data envelopment analysis approach. The paper discusses the approach in detail with a case study of a pharmaceutical company. The assessment of organizational units is another important area addressed by the DEA. Sun (2004) examines the opportunities in using the DEA as a fundamental tool for evaluation of the joint maintenance shops in the Taiwanese Army and their continuous improvement. The author discussed the DEA as a valuable benchmarking tool for the managers of the maintenance shops. The use of DEA leads to more efficient use of scare resources.

3.3 Data Envelopment Analysis (DEA)

In this section, DEA methodology and the framework of the developed DEA solver are discussed. After a brief intoduction to the DEA, the basic DEA models are explained with essential theoretical perspective which provides a fundamental background for DEA. After the basic models are introduced, an overview of existing DEA software is given. Since it is critical to know when and where DEA is an appropriate method, advantages and drawbacks of DEA are explained. In order to obtain reasonable outcomes, researchers and practitioners should bear in mind the conditions where DEA is a valid instrument. Lastly, the developed DEA solver, SmartDEA, and its testing with the real world data of ABC company are highlighted.

3.3.1 Introduction to the DEA

DEA is a nonparametric performance evaluation technique with a wide range of application areas within various disciplines. In their breakthrough study, Charnes, Cooper, and Rhodes (1978) defined DEA as “a mathematical programming model applied to observational data and a new way of obtaining empirical estimates of relations such as the production functions and/or efficient production possibility surfaces”. Since it is first introduced in 1978, it did not take long for researches to distinguish DEA as a promising tool that is easily adaptable to various application areas for performance evaluation and benchmarking of operational processes. Different entities, which are called *Decision Making Units* (DMU) in DEA terminology, can be conveniently compared in terms of multiple inputs and multiple outputs, by not requiring a priori form of relationship between inputs and outputs and fixed weights to the inputs and outputs of a DMU to get a total productivity or efficiency ratio.

DEA simply provides an *efficiency score* between 0 and 1 for each DMU involved in the analysis. The efficiency score for a DMU is determined by computing the ratio of total weighted outputs to total weighted inputs for it. DEA enables variable weights, which are calculated in such a way that the efficiency score for the DMU is maximized. (Cooper et al., 2006). For a DEA model with n different DMUs, n different linear programming models are solved to compute the efficiency scores of each of the DMUs.

A basic DEA can provide some important tools to monitor comparative performances of different entities and take managerial actions to improve them. An *efficient frontier* or *envelopment surface*, which is drawn by the "best" practicing DMUs, is the critical component of a DEA model. It is formed by the efficient DMUs which have efficiency scores of unity. The efficiency score is fundamentally the distance from each DMU to this frontier and the efficiency scores of the inefficient DMUs are calculated in accordance with this distance represented as a Pareto ratio. Besides the efficiency scores, another one is the *reference sets*, or peer groups. For each inefficient unit, DEA identifies a set of corresponding efficient units which are said to constitute this peer group for the selected DMU. The solution of the linear programming formulation of the model results in the reference set for each DMU. Knowing that the DMUs in the reference set are relatively efficient and have the same input and output structure, they can be regarded as "good" operating practices for the corresponding inefficient DMU (Boussofiane et al., 1991).

The percentages of each reference set unit contributing to the composite unit (i.e. virtual producer - with respect to which the efficiency score of the inefficient DMU is found) are also obtained by the solution of the linear programming formulation of the DEA model. One important caveat about DEA is about the rank ordering of the efficiency scores for each DMU. Since each inefficient DMU is evaluated relative to its peers, it may be misleading and theoretically, only the DMUs with the same reference sets can be strictly rank ordered (Avkıran, 1999). Another important tool DEA provides is the *target inputs* and *target outputs*, which are also named as *projections*. They represent up to which value an input should be decreased while keeping the outputs at the the same levels (i.e. input orientation) or how much an output should be increased while the input level remains unincreased (i.e. output orientation), respectively, so that the DMU becomes efficient.

3.3.2 Basic DEA Models

Since proposed by Charnes, Cooper and Rhodes (1978) in their seminal work *Measuring Efficiency of Decision Making Units*, the CCR model served as the origin of many following ideas and models in DEA literature. Cooper et al. (2006) discusses the model in detail together with the classical alternative models.

Let's suppose that there are n DMUs in the model: $\mathrm{DMU_1}, \mathrm{DMU_2}, \ldots, \mathrm{DMU_n}$. Suppose there are m inputs ans s outputs for each one of them. For $\mathrm{DMU_j}$ the inputs and outputs are represented by (x_{1j}, x_{2j}, x_{mj}) and (y_{1j}, y_{2j}, y_{sj}), respectively. As stated previously, for each $\mathrm{DMU_o}$, DEA tries to maximize the ratio

$$\frac{\text{Virtual output}}{\text{Virtual input}} \tag{3.1}$$

where

$$\text{Virtual input} = v_1 x_{1o} + \ldots + v_m x_{mo} \tag{3.2}$$

$$\text{Virtual output} = u_1 y_{1o} + \ldots + u_s y_{so} \tag{3.3}$$

and the weights v_i and u_r are not fixed in advance; in contrast, best weights are assigned according to the solution of the following fractional DEA model (**M1**):

$$\begin{aligned} (FP_o) \quad & \max \quad \theta = \frac{u_1 y_{1o} + \ldots + u_s y_{so}}{v_1 x_{1o} + \ldots + v_m x_{mo}} \\ \text{s.t.} \quad & \frac{u_1 y_{1o} + \ldots + u_s y_{so}}{v_1 x_{1o} + \ldots + v_m x_{mo}} \leq 1 \\ & v_1, v_2, \ldots, v_m \geq 0 \\ & u_1, u_2, \ldots, u_s \geq 0 \end{aligned}$$

The fractional model can be transformed into the linear model shown below (**M2**):

$$\begin{aligned} (LP_o) \quad & \max \quad \theta = u_1 y_{1o} + \ldots + u_s y_{so} \\ \text{s.t.} \quad & v_1 x_{1o} + \ldots + v_m x_{mo} = 1 \\ & u_1 y_{1j} + \ldots + u_s y_{sj} \leq v_1 x_{1j} + \ldots + v_m x_{mj} \\ & v_1, v_2, \ldots, v_m \geq 0 \\ & u_1, u_2, \ldots, u_s \geq 0 \end{aligned}$$

The fractional model (M1) is equivalent to linear model (M2), and *Unit Invariance Theorem* says that the optimal values of *max* $\theta = \theta^*$ in M1 and M2 are independent of the units in which the inputs and outputs are measured under the requirement that these units are the same for every DMU.

The input and output data can be arranged in matrix notation X and Y, respectively:

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ . & . & \cdots & . \\ . & . & \cdots & . \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

$$Y = \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ . & . & \cdots & . \\ . & . & \cdots & . \\ y_{s1} & y_{s2} & \cdots & y_{sn} \end{pmatrix}$$

Based on the input and output matrices, the linear CCR model (**M2**) can be rewritten as follows with row vector $\boldsymbol{v}$ for input multipliers and row vector $\boldsymbol{u}$ for output multipliers. This form is called *multiplier form* where multipliers $\boldsymbol{u}$ and $\boldsymbol{v}$ are treated as variables (**M3**):

$$\begin{aligned} (LP_o) \quad \max \quad & \boldsymbol{u}\boldsymbol{y_o} \\ \text{s.t.} \quad & \boldsymbol{v}\boldsymbol{x_o} = \mathbf{1} \\ & -\boldsymbol{v}X + \boldsymbol{u}Y \leq 0 \\ & \boldsymbol{v} \geq \mathbf{0} \\ & \boldsymbol{u} \geq \mathbf{0} \end{aligned}$$

The dual of model M3 which is expressed with a real variable θ and a nonnegative vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)^T$ of variables is named *envelopment form* (**M4**):

$$\begin{aligned} (DLP_o) \quad \min \quad & \theta \\ \text{s.t.} \quad & \theta\boldsymbol{x_o} - X\boldsymbol{\lambda} \geq \mathbf{0} \\ & Y\boldsymbol{\lambda} \geq \boldsymbol{y_o} \\ & \boldsymbol{\lambda} \geq \mathbf{0} \end{aligned}$$

In model M4, the objective is to guarantee at least the output level $\boldsymbol{y_o}$ of DMU$_o$ in all dimensions while reducing the input vector $\boldsymbol{x_o}$ radially (i.e. proportionally) as much as possible. Thus, M4 is referred as *CCR-Input* model.

The *slack vector* is defined as:

$$\boldsymbol{s}^- = \theta \boldsymbol{x_o} - X\boldsymbol{\lambda} \tag{3.4}$$

$$\boldsymbol{s}^+ = Y\boldsymbol{\lambda} - \boldsymbol{y_o} \tag{3.5}$$

where $s^- \in R^m$ are the input *excesses* and $\mathrm{s}^+ \in R^s$ are the output *shortfalls.*

For an inefficient DMU$_o$, its *reference set* E_o is defined as follows:

$$E_o = \{j \mid \lambda_j^* > 0\} \quad (j \in \{1, \ldots, n\}) \tag{3.6}$$

An inefficient DMU can be improved either by increasing the output values by the output shortfalls in $\boldsymbol{s}^+$ or reducing the input values in proportional with θ^* and the input excesses in $\boldsymbol{s}^-$ are removed.

That is, following *CCR Projections* formulas are applied:

$$\widehat{\boldsymbol{x_o}} \Leftarrow \theta^* \boldsymbol{x_o} - \boldsymbol{s}^- \leq \boldsymbol{x_o} \tag{3.7}$$

$$\widehat{\boldsymbol{y_o}} \Leftarrow \boldsymbol{y_o} + \boldsymbol{s}^+ \geq \boldsymbol{y_o} \tag{3.8}$$

DEA models discussed so far deal with reducing input variables while attaining at least the provided output levels. On the other hand, there exists another type of model which aims to maximize output levels while spending no more than existing resources or inputs. This kind of models are referred as *output-oriented* and *CCR-Output* model is given below (**M5**):

$$\begin{aligned} (DLPO_o) \quad \max \quad & \eta \\ \text{s.t.} \quad & \boldsymbol{x_o} - X\boldsymbol{\mu} \geq \mathbf{0} \\ & \eta \boldsymbol{y_o} - Y\boldsymbol{\mu} \leq \mathbf{0} \\ & \boldsymbol{\mu} \geq \mathbf{0} \end{aligned}$$

In DEA, θ^* represents the input reduction rate whereas η^* stands for the output enlargment rate (Cooper et al., 2006). It is clear that the less the η^* value, the more efficient the DMU, or vice versa. In order to obtain an efficiency score of between 0 and 1 and to relate the efficiency scores with the input oriented model, $1/\eta^*$ is used to express the efficiency score of the DMU in the output oriented model. Models M4 and M5 do not change the DMUs located on the efficient frontier; however target values of projections are calculated in a different way (Ulus et al., 2006).

The slack vector of the output-oriented model is defined by:

$$\begin{aligned} \boldsymbol{t}^- &= \boldsymbol{x_o} - X\boldsymbol{\mu} && (3.9)\\ \boldsymbol{t}^+ &= Y\boldsymbol{\mu} - \eta\boldsymbol{y_o} && (3.10) \end{aligned}$$

For the output oriented model, the projections or target values for input and output variables are calculated as follows:

$$\begin{aligned} \widehat{\boldsymbol{x_o}} &\Leftarrow \boldsymbol{x_o} - \boldsymbol{t}^- && (3.11)\\ \widehat{\boldsymbol{y_o}} &\Leftarrow \eta^*\boldsymbol{y_o} + \boldsymbol{t}^+ && (3.12) \end{aligned}$$

Input and output oriented CCR models which are discussed so far rely on the assumption of *constant returns to scale* (CRS) which is not a realistic case in the real world; that is, it is difficult to expect an increase/decrease in output variables proportional to an augmentation/reduction in input variables in most of the cases. Another widely known DEA model developed by Banker, Charnes and Cooper (1984) enables the problem to be *variable returns to scale* (VRS). In BCC models, production frontiers are spanned by the convex hull of the existing DMUs and these frontiers have piecewise linear and concave characteristics (Cooper et al.,2006). The BCC models differs from the previously mentioned CCR models with a convexity condition in its constraints. The linear programming model of input-oriented BCC model (*BCC-Input*) is given below (**M6**) where $\boldsymbol{e}$ is the unit vector that has 1 at all indices:

$$\begin{aligned} (BCC_o) \quad \min \quad & \theta_B \\ \text{s.t.} \quad & \theta_B\boldsymbol{x_o} - X\boldsymbol{\lambda} \geq \mathbf{0} \\ & Y\boldsymbol{\lambda} \geq \boldsymbol{y_o} \\ & \boldsymbol{e}\boldsymbol{\lambda} = 1 \\ & \boldsymbol{\lambda} \geq \mathbf{0} \end{aligned}$$

For an inefficient DMU in BCC model, its reference set, E_o is defined based on an optimal solution $\boldsymbol{\lambda}^*$ as in Eq.3.6. The projection formulas to set the target input and output variables are Eq.3.7 and Eq.3.8.

The envelopment form of the output-oriented BCC model is formulated as

$$
\begin{aligned}
(BCC - O_o) \quad \max \quad & \eta_B \\
\text{s.t.} \quad & \boldsymbol{x_o} - X\boldsymbol{\lambda} \geq \mathbf{0} \\
& \eta_B \boldsymbol{y_o} - Y\boldsymbol{\lambda} \leq \mathbf{0} \\
& \boldsymbol{e\lambda} = 1 \\
& \boldsymbol{\lambda} \geq \mathbf{0}
\end{aligned}
$$

BCC versions of the Eq.3.11 and Eq.3.12 are used to calculate the projection values of the input and output variables of an inefficient DMU in the BCC-Output model.

In addition to the characteristics of each model and the preference of input or output orientation, the signs of input and output data are also taken into account in the selection of the model that will be used in an analysis. Table 3.1 summarizes the requirements on the signs of data that will be included in the model. *Semi-p* stands for semipositive data. Semipositive data means a set of nonnegative data points in which at least one of them is positive. For instance, consider a variable that is included as an input in the model. It may take a nonnegative value (i.e. either zero or a greater value), but at least for one DMU it has to take a positive value. *Free* allows the use of either negative, positive or zero data.

Table 3.1: Characteristics of cases handled by DSS

Model Data	CCR-I	CCR-O	BCC-I	BCC-O
X	Semi-p	Semi-p	Semi-p	Free
Y	Free	Free	Free	Semi-p

3.3.3 Advantages and Disadvantages of DEA

DEA provides significant number of opportunities in the performance evaluation of entities and their benchmarking. Increasing popularity in the literature and the boom in the number of applied DEA studies in last decades show the extent of attention paid by the researchers. However, it is important to remember that DEA can offer reasonable solutions and guidelines only if it is used wisely under appropriate settings. Thus, disadvantages and drawbacks of DEA methodology should be clearly perceived before any prospective analysis.

Let's first summarize the advantages of DEA possesses. First of all, DEA can deal with multiple inputs and outputs without a priori relationship among them. In addition, the units of inputs and outputs do not need to be congruent. Cooper et al. (2006) discuss additional advantages of DEA such as its ability to indicate sources and amounts of inefficiency for each input and output variable belonging to a DMU. In terms of computational requirments, DEA models do not need high performance solvers unless the size of the problem is very large.

Besides the advantages, DEA has the following drawbacks that are common in many models. Availability and reliability of the data needed in the analysis may be a serious problem (Easton et. al., 2002) and missing data points in inputs or outputs of a DMU can lead to its exclusion from the model. DEA is strongly affected by possible errors and extreme points in data (Smith et al., 2002). In addition, DEA can only measure *relative* efficiency. It can not lead to a rank ordering of DMUs by comparing their efficiencies with an absolute theoretical efficiency value. Moreover, even if a DMU emerges with a position on the efficient frontier, it may not perform well in real life. Since it is superior only in a specific observation set of DMUs, the best way to overcome this obstacle is to keep the number of DMUs contained in the analysis large. As a rule of thumb, the number of DMUs in the model should exceed the sum of number of inputs and outputs several times.

3.3.4 An Overview on DEA Software

As a linear programming technique, DEA mainly relies on the solution techniques of linear programming models. Availability of a wide range of linear programming solvers make DEA an affordable and convenient method to measure efficiencies of entities quantatively. In addition, as the application areas of DEA expand and the methodology find more attention not only from academicans but also from practitioners, the extent and quality of DEA software available in the market prominently increased.

Hollingsworth (1997) and Bowlin (1998) reviewed three DEA softare package: Warwick DEA for Windows, IDEAS and Frontier Analyst. The former author discussed these three software in detail in terms of the intended use and area of application, ease of use and package facilities. However, one of the most comprehensive DEA software review paper belongs to Barr (2004). The author surveyed approximately 20 DEA software packages and in the study eight of them were examined in detail together with a comparison scheme. Barr evaluated these eight DEA software - DEA Solver Pro, Frontier Analyst, OnFront, Warwick DEA, DEA Excel Solver, DEAP, EMS, and Pioneer according to eight primary criteria:

1. Available models the software offers
2. Key DEA features and capabilities such as non-discretionary or categorical factors, priorities on variables and sensitivity analysis
3. Platform and interoperability
4. Existence of user interface
5. Reporting such as seperate worksheets, customized reports, graphs and charts,
6. Documentation and support such as availability of technical reference manual with modeling details, availability of user guide, technical customer support
7. Test performance
8. Availability (i.e. academic and commercial licencing costs, maintenance cost, and availability of free demo)

Barr divided the eight software into two subgroups: Commercial (EA Solver Pro, Frontier Analyst, OnFront, Warwick DEA) and Noncommercial (DEA Excel Solver, DEAP, EMS, Pioneer). Then, the author gave the details of all these software package in a comparison chart that is completed according to the fulfillment of the criteria cited above.

3.4 Integration of DEA Results with Data Mining and Information Visualization

3.4.1 Extensions in the Analysis of DEA Results

The meaning of DEA results needs interpretation for the transformation of mathematical terms into managerial insights to assess and improve the performances of inefficient DMUs. Significant amount of data that come by DEA results are open to further detailed analysis for the derivation of interesting insights and guidelines. Many of the data mining and information visualization techniques are very effective tools for this analysis. Representation of DEA results in accordance with on-line analytical processing (OLAP) technology can enable the managers and analysts to involve in faster and better decision making processes by performing multidimensional analysis of DEA results data.

Keim (2002) and Spence (2001) discusses the information visualization as one of the important growing field of computer science, combining computer graphics, data mining activities and explanatory data anaysis to better understand the data in a visual perspective. DEA solutions include efficiency scores, reference sets for each DMU and projection values for the input and output variables belonging to a DMU. They provide important in-depth information about the system; however, some hidden patterns or important insights in the DEA results data may remain undiscovered by not using the benefits of state-of-the-art visualization techniques in data analysis.

Ulus et al. (2006) discusses the data visualization as a fundamental concept of data analysis, helping the analyst with detecting outliers, discovering underlying patterns which are not possible to recognise with classical visualization techniques in statistics (i.e. histograms, quantile plots, box plots, symmetry plots etc.), and coming up with new insights and hypotheses. In the paper, authors used *colored scatter plot* to help building insights in a benchmarking study of industrial transportation companies traded in the New York Stock Exchange (NYSE). In the study, DEA is used as the primary methodology as the selected financial data of the companies are used as the input and output variables in the DEA model.

Ertek et al. (2007) presents unobserved patterns in the benchmarking of Turkish apparel industry by the help of information visualization techniques. In the paper, DEA is applied to determine the efficiency scores of the companies. Inputs, outputs, other related data and efficiency scores are visualized in *colored scatter plots* and *tile graphs* by means of Miner3D (Miner3D) and Visokio Omniscope (Omniscope) software, respectively. Former is also referred as *starfield visualization* in information visualization terminology. The x-axis and y-axis of the starfield can represent various variables of the data on a two dimensional physical space. Similarly, color can be attributed to a set of variables and interesting insights or underlying patterns are seeked in the visualization. The latter visualization type, tile graph, basicly divides a bounded surface according to a given criteria by allocating appropriate area for each visualized entity. A carefully designed coloring scheme can also contribute to the effectiveness of the tile graph as well.

Effective and creative data visualizations can significantly reduce the time to understand the information and patterns conveyed with the data as well as contributing to the generation of insights about them. They also promote the formation of guidelines about the investigation of the data. Detailed reviews and information about data visualization terminology and applications can be found in the review papers by Keim (2002) and Hoffman and Grinstein (2002) and in the books by Soukup and Davidson (2002) and Spence (2001).

3.4.2 Framework for the Integration of Data Mining and Information Visualization Techniques with DEA Results

In this section, we propose a convenient notation to represent the given DEA data and the model solutions in a formal manner. This notation will be used to describe the framework for the generation and representation of DEA results such that this framework allows further data mining and information visualization activities. The notation is built in accordance with the widely used DEA notation available in the literature.

In the first part of the framework the notation is given in terms constants, indices, sets, DMU attributes, DMUs (i.e. representing the DMUs as objects), DEA results data and functions.

CONSTANTS

D : Number of distinct DEA models

N : Number of DMUs that exist in at least one of the DEA models

n_d : Number of DMUs in model d

K : Number of attribute groups

A_k : Number of atributes in attribute group k

A : Number of attributes

m_d : Number of inputs in model d

s_d : Number of outputs in model d

τ_d : Number of other data in model d

INDICES

d : Model

j : DMU index

h : DMU index in a reference set

k : Attribute group

l : Index in the attribute group k

a : Attribute index

i : Input index

r : Output index

t : Other data index

DMU ATTRIBUTES

α : The attribute object α

α_{kl} : l^{th} attribute of the k^{th} attribute group

α_a : a^{th} attribute

SETS

$\mathcal{D}$: Set of all DEA models

$\mathcal{N}$: Set of all DMUs

$\mathcal{N}_d$: Set of DMUs in model d

$\mathcal{I}$: Set of all inputs

$\mathcal{I}_d$: Set of inputs in model d

$\mathcal{R}$: Set of all outputs

$\mathcal{R}_d$: Set of outputs in model d

$\mathcal{T}$: Set of all other data

$\mathcal{T}_d$: Set of other data in model d

$\mathcal{A}$: Set of all attributes

$\mathcal{A}_k$: Set of all attributes in attribute group k

$\mathcal{A} = \cup_{\forall k}\mathcal{A}_k$, (according to their semantic meaning)

$\mathcal{A} = \mathcal{I} \cup \mathcal{R} \cup \mathcal{T}$, (according to their roles in the model)

DMUs

ψ_j : DMU j

Ψ_j : DMUs in the reference set of DMU j

FUNCTIONS

$$\Gamma_m(\alpha) \quad : \quad \begin{cases} 1 & \text{if} \quad \alpha \in \mathcal{I} \\ 2 & \text{if} \quad \alpha \in \mathcal{R} \\ 3 & \text{if} \quad \alpha \in \mathcal{T} \end{cases}$$

$i(\alpha)$: the input index of attribute α in the model given that the attribute is an input

$r(\alpha)$: the output index of attribute α in the model given that the attribute is an output

$$ir(\alpha) \quad : \quad \begin{cases} i(\alpha) & \text{if} \quad \Gamma(\alpha) = 1 \\ r(\alpha) & \text{if} \quad \Gamma(\alpha) = 2 \\ \text{Error} & \text{if} \quad \Gamma(\alpha) = 3 \end{cases}$$

DEA RESULTS DATA

θ_{jd}	:	Efficiency of j^{th} DMU in d^{th} model
E_{jd}	:	Reference set of the j^{th} DMU in d^{th} model
λ_{hjd}	:	Reference weight of the h^{th} DMU in the reference set E_j in model d
x_{ijd}	:	Original value of input i for j^{th} DMU in model d
$\boldsymbol{x_{jd}}$	:	Original input vector for j^{th} DMU in model d
$\widehat{x}_{ijd}$	:	Projection value of input i for j^{th} DMU in modeld
$\widehat{\boldsymbol{x}}_{\boldsymbol{jd}}$	:	Projection input vector for j^{th} DMU in modeld
y_{rjd}	:	Original value of output r for j^{th} DMU in model d
$\boldsymbol{y_{jd}}$	:	Original output vector for j^{th} DMU in model d
$\widehat{y}_{rjd}$	:	Projection value of output r for j^{th} DMU in model d
$\widehat{\boldsymbol{y}}_{\boldsymbol{jd}}$	:	Projection output vector for j^{th} DMU in model d
z_{tjd}	:	Value of other data t for j^{th} DMU in model d
$\boldsymbol{z_{jd}}$	:	Other data vector for j^{th} DMU in model d

If the model under consideration is known or there is just one DEA model, d can be dropped from the formulas.

Table 3.2: Output Format 1: Efficiency Scores

Field Index	**Field Description**	**Notation in Framework**
1	DMU Name	ψ_j
2	Efficiency Score	θ_j
Next m fields		$\boldsymbol{x_j}$
Next s fields		$\boldsymbol{y_j}$
Next $\tau fields$		$\boldsymbol{z_j}$

Table 3.2 gives the proposed structure of output data for efficiency scores. This DEA results format constitutes first part of the framework which is composed of three such data representation table. As noticed, a database point of view and notation are considered. In Table 3.2, a record (row) exists for each element of the set $\{j : j \in \mathcal{N}\}$ and the number of records is N.

Table 3.3: Output Format 2: Reference Sets

Field Index	**Field Description**	**Notation in Framework**
1	DMU Name	ψ_j
2	Efficiency Score	θ_j
3	h^{th} DMU in the reference set E_j of DMU$_j$	Ψ_{hj}
4	Reference weight of h^{th} DMU in the reference set E_j of DMU$_j$	λ_{hj}

Table 3.4: Output Format 3: Projections

Field Index	**Field Description**	**Notation in Framework**
1	DMU Name	ψ_j
2	Efficiency Score	θ_j
3	Attribute Name	θ
4	Is input or output	$\Gamma(\alpha)$
5	Original value	$x_{ir(\alpha),j}$
6	Projection nalue	$\widehat{x}_{ir(\alpha),j}$
7	Difference	$x_{ir(\alpha),j} - \widehat{x}_{ir(\alpha),j}$
8	Percentage Difference	$100(x_{ir(\alpha),j} - \widehat{x}_{ir(\alpha),j})/x_{ir(\alpha),j}$

Table 3.3 constitutes the second part of the framework. It describes how the reference sets are represented in the framework.

In Table 3.3, a record (row) exists for each element of the set $\{(j, h) : j \in \mathcal{N}, R \in E_j\}$ and the number of records is $\sum_{j \in \mathcal{N}} |E_j|$.

Table 3.4 constitutes the last part of the framework. It describes how the projection variables are represented in the framework. In Table 3.4, a record (row) exists for each element of the set $\{(j, \alpha) : j \in \mathcal{N}, \alpha \in (\mathcal{N} \cup \mathcal{R})\}$ and the number of records is $N \times R$. In the next section, the implementation of this framework is discussed in a real world application by using a novel DEA solver.

3.5 The Developed Software: SmartDEA

After the advantages and disadvantages of the various state-of-the-art DEA Solvers are carefully reviewed, it is decided to build a completely new DEA software considering the following facts:

1. The open source and commercial DEA software in the market lack an effective model data and solution representation in their output files. Thus, most of the time further analysis using theses files can be a cumbersome task to arrange the files for information visulization and data mining.

2. The developed software must be compatible with the proposed framework. In order to reduce the efforts and save time in analyzing the results of the DEA, an effective and ready-to-import output file generation function is needed as suggested in the framework.

3. Commercial DEA software in the market provides high performance but with a serious licencing cost ranging from £800 to £2500.

4. Freely available solvers exist but with a limited number of DMUs which restricts the analysis we plan to conduct. It is a fact that the limit on DMUs with the free software can be too limiting for a real life application which can have hundreds DMUs.

5. The Graphical User Interface (GUI) design and development is commonly neglected in these software while it is a critical feature for the use of the sofware in industrial applications by theoretically incompetent practitioners.

6. An innovative, effective and user-friendly software can lead to advancements in future DEA research with different data in other problem scenarios.

The coding of the decision support system is completed in the C# language under MS Visual Studio.NET 2005. The developed DEA software is named as SmartDEA Solver. As the mathematical programming model solver, a free and publicly available lp_solve dynamic link library (.dll) file is embedded in the software and called whenever the linear programming model is solved. That is, optimization task of the software is outsourced to lp_solve library. As stated before, integration of the DEA results with information visualization and data mining tools is one of the main objectives of the study. In accordance with this goal, we give special importance in the format of the output file. We devised the output file format as MS Excel (.XLS) file. The widespread use of Microsoft Office packages in Windows platforms and the fact that any user can conveniently create, modify and store data in this file format let the data import and export with SmartDEA solver be very rapid and in a user-friendly environment. The data format written in the output file is designed to visualize and mine the data effectively without further data format modification and cleaning efforts.

There are five main stages in the DEA solution process which is summarized on the first window that appears on the screen after initialization of the software (Figure 3.1). User selects the data file in MS Excel format to import the DEA data into the SmartDEA solver. After the file is chosen in the file dialog box, second window is appeared on the screen, asking the spreadsheet that will be used to construct the DEA model. That is, the developed software supports MS Excel files with multiple spreadsheets; users can place different data into different spreadsheets and they can be separately loaded into the software with this mechanism. In addition, there exists a control about the consistency of the data in the selected spreadsheet. The data should be in the required format, with each of the DMU Name, inputs, outputs and other data fields are arranged in columns. Each row represents the variables for a DMU and no space should be added between any columns on the spreadsheet. If the selected spreadsheet is not in the required format controlled by the consistency check, an error message appears and user is asked to select any other spreadsheet. Figure 3.2 shows the spreadsheet selection window.

After a spreadsheet is selected, if the data on the spreadsheet is in the required format, the third window appears on the screen (Figure 3.3) in which the user contructs the DEA model. There are three buttons in the model selection group box, representing the DMU names, inputs and outputs. After the user selects the appropriate field in the listbox, it is selected as a DMU name field, input field or output field by clicking on these buttons. There is no need to follow any order

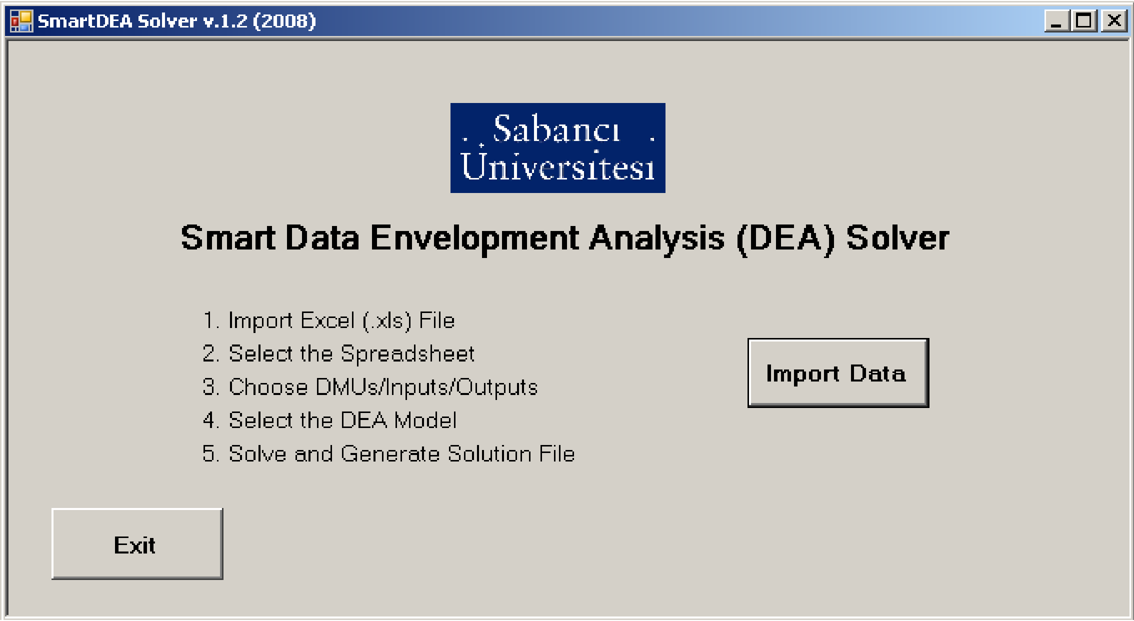

Figure 3.1: First window that appears after the initialization of the software

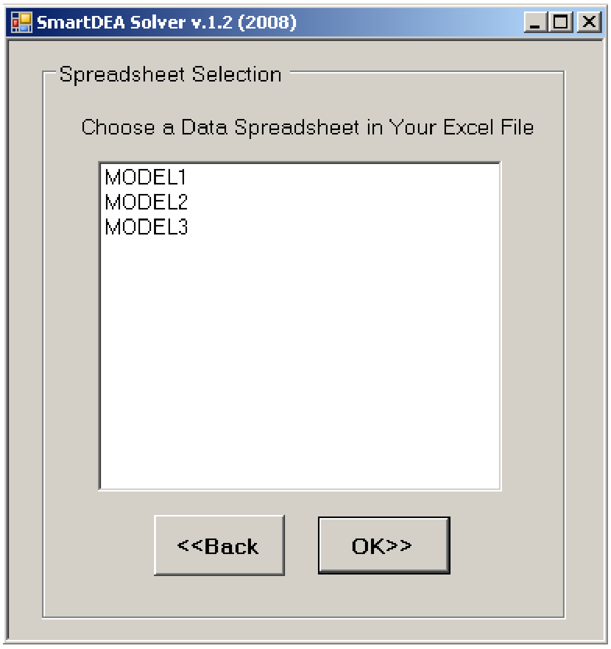

Figure 3.2: Spreadsheet selection window

in model construction but at least one input and one output is required as well as exactly one DMU name field. The reset operation can be performed on any previously selected field by using either Clear Selected Field and Clear All buttons. Any selected field is distinguished from other fields by a coloring scheme. A selected DMU name is colored light purple, while yellow-green and orange colors are reserved for inputs and outputs in the model. Other data that come within the spreadsheet are not colored, distinguishing them from the model data.

The next stage in the SmartDEA solver is the selection of the model type. Classical DEA models, specifically CCR-Input, CCR-Output, BCC-Input and BCC-Output, are available in the solver. The user chooses one of these model types by selecting it in the listbox and clicking on the OK button (Figure 3.4). Depending on the size of the model, but generally in the order of miliseconds, SmartDEA solver returns the efficiency scores, reference set and projection values for each DMU in the model. DEA solution summary is given on the left side of the solution display window of the software (Figure 3.5). The Efficiency score of each DMU is provided with a self-explanatory coloring scheme which displays efficient DMUs light sky blue and an inefficient DMU in blue. DMU detail group box on the right side of the window provides all information about the selected DMU. In addition to the reference set and reference weight information, target values or projections of each input and output variable are given at the end of the grid view. Absolute and percentage differences between the original and target values of input and output variables are also displayed on the window. The user can access all this information about a DMU either by selecting it in the combobox and displaying them on the window or by generating a report file in the MS Excel format.

Once the user clicks on the Generate Report button, an .XLS file is formed with three distinct spreadsheets named as *Efficiency Scores*, *Reference Sets* and *Projections*. Each one of them stores the output data of the SmartDEA solver in a database format, that is each variable is in different columns with each row represents a unique information about the solution. The data structure of each generated spreadsheet is given in Table 3.5, Table 3.6 and Table 3.7, respectively. Field names and descriptions of them are provided in these tables. The outstanding feature of these field structures is its ability to enable the user quickly analyze the

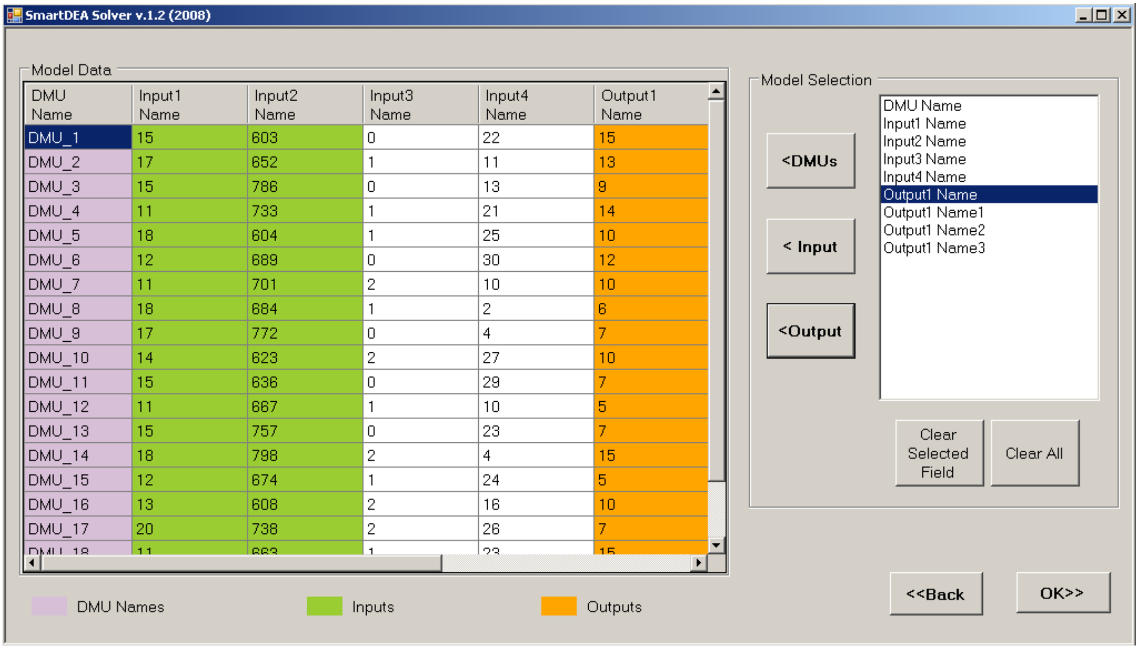

Figure 3.3: Model construction window

Table 3.5: Data structure of "Efficiency Scores" spreadsheet

Field No	Field Description
1	DMU Names
2	Efficiency Scores
Next m+s+τ	Model Data (Inputs, outputs and other data are highlighted)

DEA solution and they make data import process into visualization tools less time consuming and more convenient.

In order to deploy the software into any computer easily, a setup package is built, including the required MS office library files, executable application file and lp_solve library file. The user can install the SmartDEA solver into any computer with the option of personal and public use. The application file directory can be changed during the installation process. Since the software is developed under .NET environment, the computer into which the SmartDEA is deployed needs .NET Framework 2.0 installed to be able to run the software.

As noticed in the stages of solution process described above, users move to the next stage progressively after the previous one is completed appropriately. It is

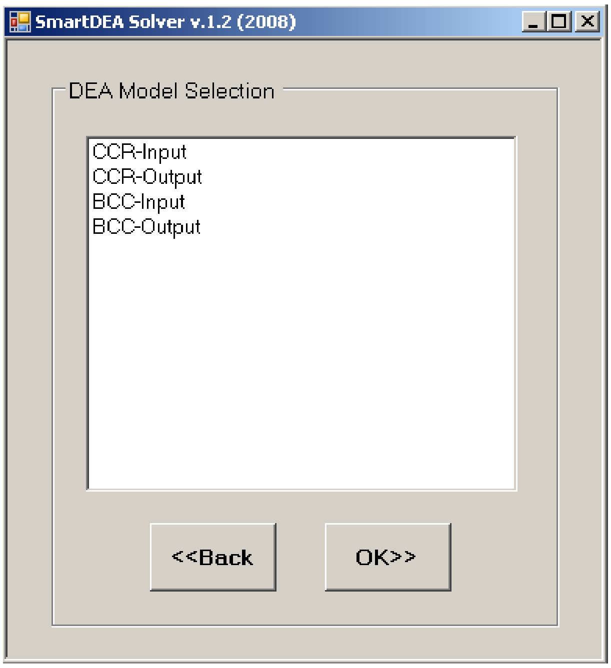

Figure 3.4: DEA model type selection window

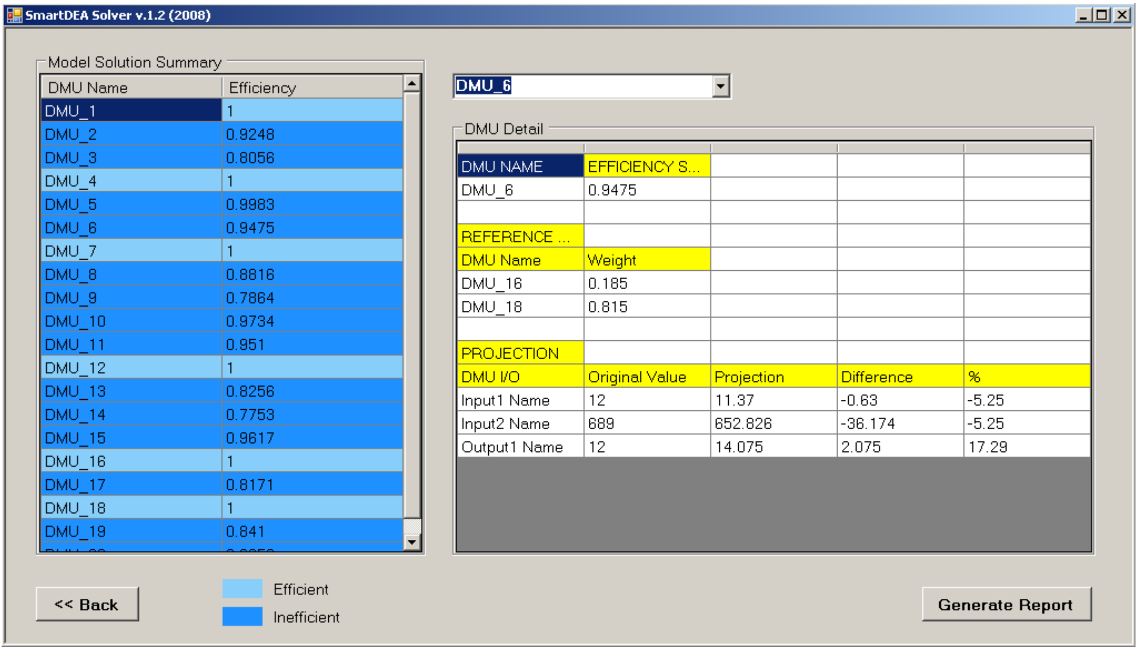

Figure 3.5: Solution display window of the software

Table 3.6: Data structure of "Reference Sets" spreadsheet

Field No	Field Description
1	DMU Names
2	Efficiency Scores
3	Reference DMU_i
4	λ_i

Table 3.7: Data structure of "Projections" spreadsheet

Field No	Field Description
1	DMU Names
2	Efficiency Scores
3	Variable Names
4	Input or Output
5	Original Value
7	Difference
8	Percentage Difference

also available for users to go back one step and do the current activities again with different settings. This capability saves time and brings convenience especially when users work with same model data but different input or output settings or different models. For instance, let a user work with the transportation activities data of a logistics company. But he or she is not sure about the correct selection of inputs and outputs or about the DEA model such as BCC-Input or CCR-Input. In such a case, the structure of the software enables the user carrying out different analysis without loading the model data again and again.

Figure 3.6 gives the Unified Modelling Language (UML) activity diagram of the SmartDEA solver. Activity diagrams are mainly used for operational process modeling. In addition to being a visual modeling tool, it also helps software developers to convey and document their ideas and work in a systematic and effective manner. UML activity diagram given in Figure 3.6 clearly express the software structure independent from the programming language and it helps other developers undestand the logic and the structure, leading to a continuous development with multiple developers. The activity diagram effectively summarizes what happens after the initialization of the software.

Compared to other DEA solvers in the market, the advantages of SmartDEA is its effective output file data structure especially suitable for visualization tools such as Miner3D (Miner3D) and Visokio Omniscope (Omniscope). A further step in the development of the software can be the integration with any visualization and data mining tool as endorsed by the suggested software framework.

3.5.1 Test of the SmartDEA software: Benchmarking Dealers in a Turkish Automotive Company

The continuous performance evaluation is a critical management objective in today's highly competitive markets. In order to be succesful in the fierce market conditions, companies need to measure the performances of themselves and their business partners to make any improvement in the existing conditions. In this section, the test of the SmartDEA is made by using two different DEA models for benchmarking dealers of a leading Turkish automotive company, ABC. In these models the dealers are benchmarked firstly in terms of their overall operations and then in terms of

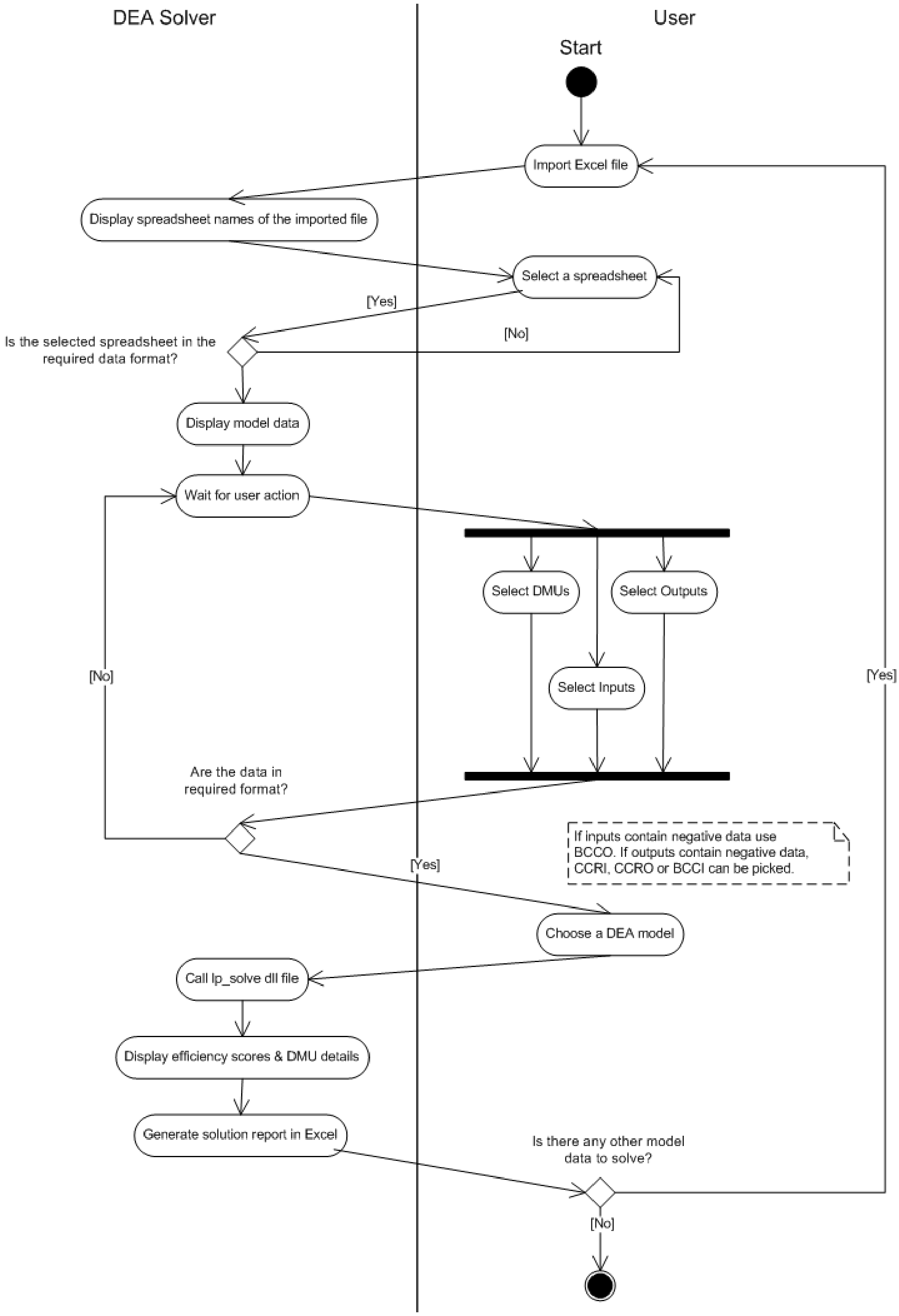

Figure 3.6: UML activity diagram of the SmartDEA solver

their purchases from ABC, respectively. The relationships of the ABC's spare parts warehouse and its dealers are analyzed by considering three inputs and one output. For both of the models, total *spare parts area* of dealers, total dealer *expenses* and *spare parts employees* of dealers are taken as inputs. However, while *total revenue* of dealers is taken as output in the first model, the amount of dealers' *purchase from ABC* is considered as the output variable for the latter model.

In this benchmarking study, BCC-Input is selected as the DEA model that will be applied in the problem. There are two primary reasons behind this choice. (1) BCC model imposes more flexibility in such a way that the solution is not restricted with constant returns to scale. As previously stated, BCC model implies variable returns to scale and it is preferable in our problem. (2) Input oriented models set targets for input variables, up to which value they can be reduced while yielding at least the same amount of outputs. We believe that building short term road maps to reduce resource consumption of DMUs is set as one of the objectives and input oriented approach enables us to get insights by looking at the projected input variables. The data used in the analysis meets the sign requirements imposed by the BCC-I model in Table 3.1. Due to confidentiality issues, in this paper we do not present the analysis, results and insights. Rather the developed DEA solver and its framework are focused as well as the data mining perspective and integration of DEA solutions into this framework.

3.5.2 Analysis Examples

In this section, applications of some of the possible data mining and information visualization techniques are exemplified. The proposed software framework provides the structure of the data used in these techniques. In order to familiarize the reader with these techniques, sample applications are discussed below. The visualizations belong to the solution of the first DEA model that is used in the benchmarking of ABC dealers.

Starfield visualization is illustrated in Figure 3.7 which is generated in the information visualization software, Miner3D (Miner3D). It is fundamentally a two-dimensional scatter plot with a capability of visualization high-dimensional data. In this visualization sample, x-axis and y-axis show the total revenue and the num-

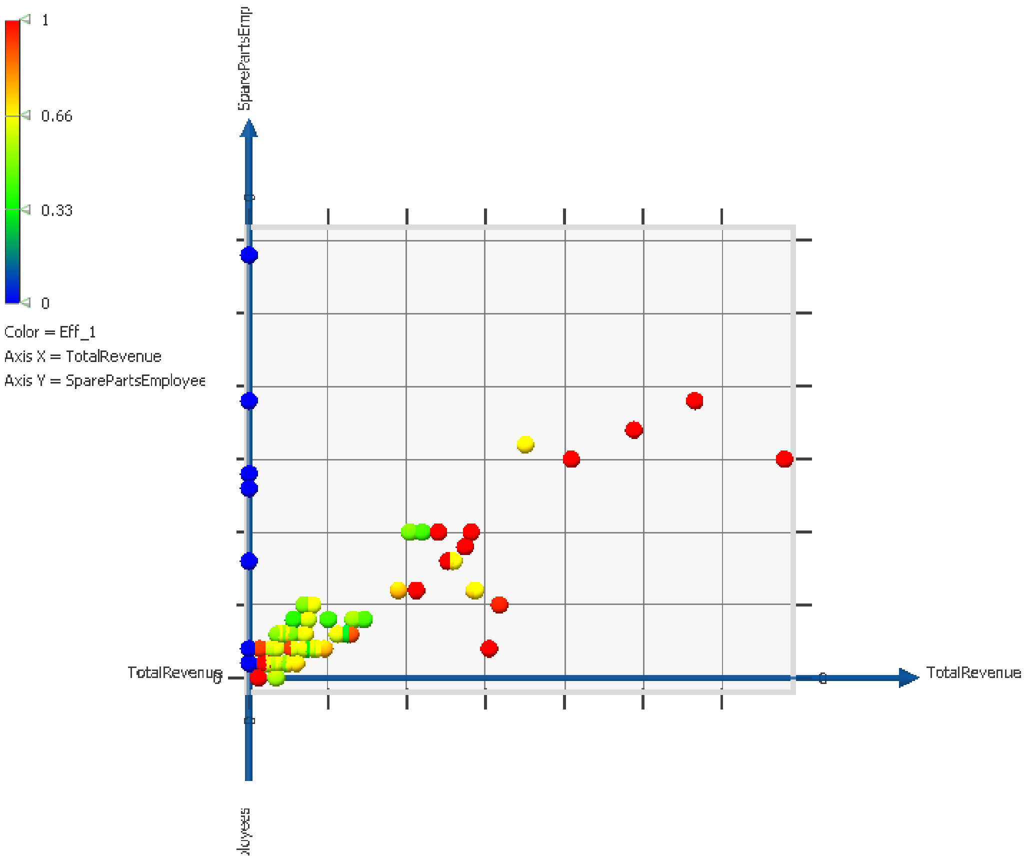

Figure 3.7: An example of starfield visualization

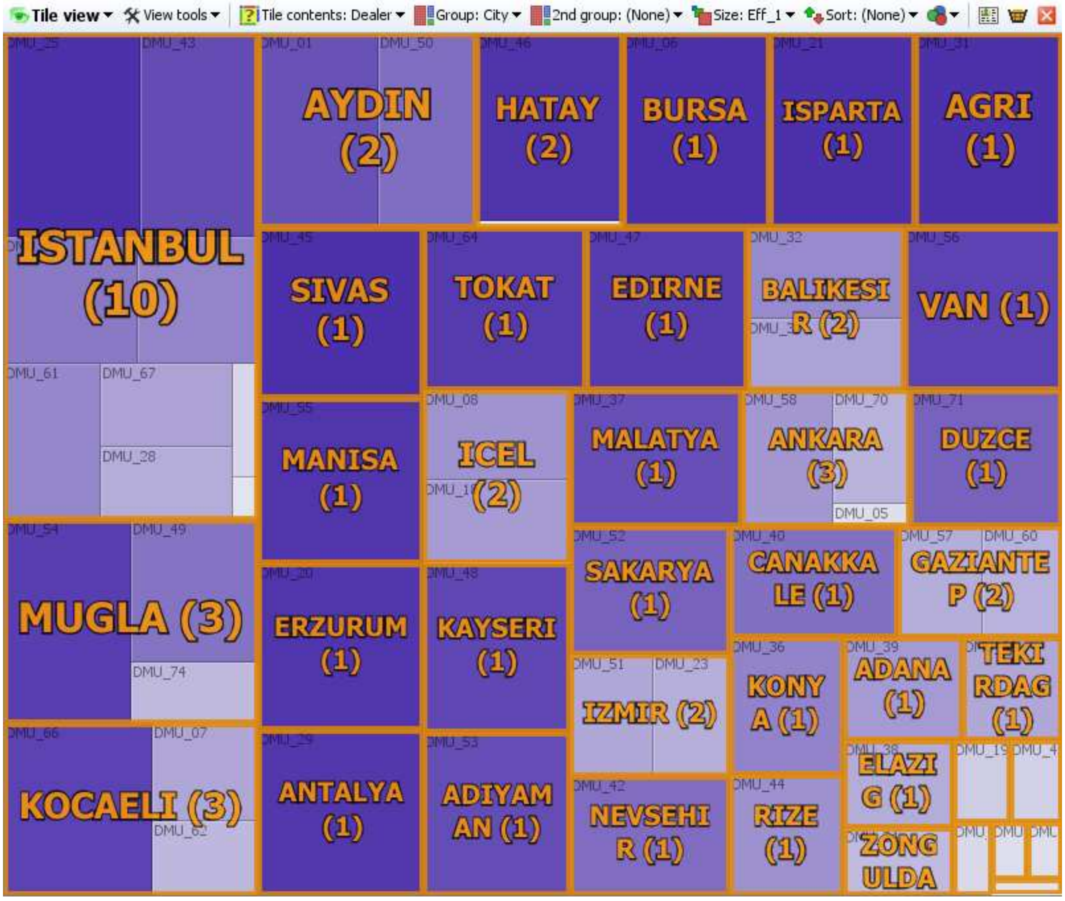

Figure 3.8: An example of tile graph

ber of spare parts employees for the dealers. The corresponding x and y values are omitted from the graph due to confidentiality reasons. The color stands for the efficiency scores of the dealers. In addition to these variables, size of the DMUs can also be assigned any other data. Important patterns can be revealed by a clever choice of four visualization parameters: *x-axis*, *y-axis*, *color* and *size*. An analytic framework makes it possible to investigate these visulization options and catch the significant insights systematically.

Another effective visualizing scheme for DEA problems are tile graphs. Figure 3.8 gives an example of tile graph generated by Visokio Omniscope software (Omniscope). It allows analysts to cluster the data according to a criterion and effectively investigate the patterns hidden in the DEA results. Similarly, analysts need to choose variables among DEA model data, other data and DEA results to visualize on the graph. A systematic framework again makes it very convenient to derive insights and generate guidelines to be followed in the management of DMUs for better performance. In Figure 3.8, tile contents are the DMUs to be benchmarked. In this sample illustration, they are grouped according to their geographical region while size and color show the efficiency values that come up with the DEA results. Due to confidentiality, randomly generated efficiency scores are used in the graph. Increasing the speed and accuracy of visual analysis, an analytical benchmarking framework can provide reliable road maps for better integration of DEA results and tile graphs. The analysts choose among the DEA model data, other data and the DEA results to be represented by the *tile content, first clustering criteria, second clustering criteria, size* and *color*.

3.6 Conclusion & Future Work

In this study, a software framework is proposed to represent the results of any DEA study in a formal manner, enabling the analysts to make analytical benchmarking between different DMUs. In accordance with this framework, an innovative DEA solver, SmartDEA, is developed and tested in a real world project for benchmarking the dealers of a Turkish automotive company. The framework allows analysts to identify hidden patterns and derive managerial insights by integrating the results of any DEA study with various types of information visualization, data mining and online analytical processing (OLAP) technologies in the implementations of DEA studies. To summarize, DEA results are examined in a computer science oriented perspective and data mining point of view.

This study provides a strong fundamental background to start any kind of analytical benchmarking in DEA by developing a formal way of representation for DEA results. One of the future work can be to devise and develop an analytical analysis framework, taking the proposed framework in this paper as a basis. Associated with this analysis framework, the ABC's DEA results can be analyzed by integrating them with data mining and information visualization techniques analytically. In addition, the software framework can be extended to allow many sub-categories of these techniques. As noticed, the proposed framework can be applied in any application area. Another future work can be focusing on any specific area and building domain specific frameworks to analyze DEA results. The developed software can be improved by adding help, information or tips for those not related with DEA concepts.

Chapter 4

Conclusion

Two stand-alone problems are discussed in this thesis with a common perspective: Building software frameworks for the effective solution of each problem type. The addressed topics are selected from (1) production scheduling, and (2) the integration of DEA with information visualization and data mining techniques. Innovative software are designed and built in order to solve the selected problems which are encountered in real world applications. The software discussed in the first chapter is publicly available with its source code and documentation.

The first topic is an operational level planning problem that widely arises in chemicals industry. The effective product sequencing is achieved with an innovative, user-friendly software which also enables the planners to group products, process urgent orders, record production history and visualize the planned product sequence in a single mixer. The software converts each different production scenario into a single TSP structure which allows the use a single solution technique for all production setting. The product sequencing software is implemented at the production faclilities of a global manufacturer of the hygiene and chemical products.

At the second part of the thesis, a formal representation of DEA results is proposed for the integration of data mining and information visualization techniques. The framework enables executives, managers and researchers to make analytical analysis and benchmarking between decision making units included in the DEA model. Associated with this framework, a novel DEA solver is designed and developed. The software is tested in the benchmarking of the dealers affiliated with a leading Turkish automotive company.

It is believed that both of the practical solutions, frameworks and software developed for the problems mentioned in the thesis can serve as effective management tools in many other real world applications.

Bibliography

Chapter 2

Aarts, E., Lenstra, J.K, 1997. Local Search in Combinatorial Optimization, E. H. L. John Wiley and Sons, 1st edition.

Abou Najm, M., El-Fadel, M., 2004. Computer-based interface for an integrated solid waste management optimization model. Environmental Modelling and Software, vol.19, no.12, 1151-1164.

Adonyi, R., Biros, G., Holczinger, T. and Friedler, F., 2008. Effective scheduling of a large-scale paint production system, Journal of Cleaner Production 16 (2), pp. 225232.

Almató, M., Sanmart, E., Espuna A., Puigjaner, L., 1997. Rationalizing the water use in the batch process industry. Comput Chem Eng 21: S971-S976

Almató, M., Espuna, A., Puigjaner, L., 1999. Optimization of water use in batch process industries. Comp Chem Eng 23: 1427-1437

Berlin, J., Sonesson, U., Tillman, A-M., 2007. A life cycle based method to minimise environmental impact of dairy production through product sequencing. Journal of Cleaner Production; 15: 347-356.

Bramel, J., Simchi-Levi, D., 1997. The logic of logistics: theory, algorithms, and applications for logistics management, Springer,1st edition.

Chalhoub:
http://www.codeproject.com/useritems/SimulatedAnnealing.asp?print=true

Denzer, R., 2005. Generic integration of environmental decision support systems state-of-the-art, Environmental Modelling and Software 20, pp. 1217-1223.

Gutin, G., Punnen, A.P., 2002. The traveling salesman problem and its variations, Springer, 1st edition.

Grau, R., Espuna, A., Puigjaner, L. ,1995. Environmental considerations in batch production scheduling, Computers and Chemical Engineering (Suppl. 19), pp. 651656.

Korte, B., Vygen, J., 2005. Combinatorial optimization: theory and algorithms, Springer, 3rd edition.

JohnsonDiversey:
http://www.johnsondiversey.com/Cultures/tr-TR

Matthies, M., Giupponi, C., Ostendorf, B., 2007. Environmental decision support systems: Current issues, methods and tools, Environmental Modelling and Software, vol.22, no.2, 123-127.

Muzy, A., Innocenti, E., Aiello A., Santucci, J. F., Santoni P., Hill, D., 2005. Modelling and simulation of ecological propagation processes: Application to fire spread. Environmental Modelling and Software, vol. 20, 827-842.

OMG Unified Modeling Language Specification, 2003. Version 1.5, pg. 3-155, Object Management Group, Inc.

Park, J.H., Asada, H., 1994. Sequence optimization for high speed robotic assembly using simulated annealing, Robotics and Automation, Proceedings 1994 IEEE International Conference.

Papagjorji, P., Shatar, T., 2004. Using the Unified Modeling Language to develop soil water balance and irrigation-scheduling models. Env. Mod. Soft., 19, 451-459.

Pizzolato, N.D., Canen A.G., 1998. Case study: improving industrial competitiveness: a TSP approach, Logistics Information Management, vol.11, no.3, 181-191.

Rivas, A., Irizar, I., Ayesa, E., 2008. Model-based optimisation of Wastewater Treatment Plants design, Environmental Modelling and Software, vol.23, no.4, 435-450.

Schmuller, J., 2001. Sams Teach Yourself Uml in 24 Hours, Sams, Indianapolis, IN.

Sundararajan, S., Srinivasan, G., Staehle, W.O., Zimmers E.W., 1998. Application of a decision support system for operational decisions, Computers & Industrial Engineering 35 (1-2) , pp. 141-144.

Wang, Y.P., Smith, R., 1994. Wastewater minimisation, Chem. Eng. Sci. 49 , pp. 9811006.

Toth, P.,Vigo, D., 2001. The vehicle routing problem Society for Industrial & Applied Mathematics ,1st edition.

Tsai, T. and Sato, R., 2004. A UML model of agile production planning and control system. Comp. Indust., 53, 133152.

Winston, W., 1994. Operations Research-Applications and Algorithms, Wadsworth Inc., USA

Chapter 3

Avkıran, N., 1999. An Application Reference for Data Envelopment Analysis in Branch Banking: Helping the Novice Researcher, International Journal of Bank Marketing, 17(5), 206-220.

Banker, R. D., Charnes, A., and Cooper, W.W., 1984. Some models for estimating technical and scale inefficiencies in data envelopment analysis, Management Science 30-9, 1078-1092.

Barr, R. S., 2004. DEA software tools and technology: A state-of-the-art survey, Handbook on Data Envelopment Analysis. Hingham, MA, USA: Kluwer Academic Publishers, 539.

Boussofiane, A., Dyson, R.G. and Thanassoulis, E., 1991. Applied data envelopment analysis. European Journal of Operational Research 52, pp. 1-15.

Bowlin, W. F.,1998. Measuring performance: an introduction to data envelopment analysis (DEA), Journal of Cost Analysis, 3(1), 328.

Charnes, A., Cooper, W. W., Rhodes, E., 1978. Measuring the efficiency of decision making units, European Journal of Operational Research 2, pp. 429-444.

Cooper, W. W., Seiford L.,M., Tone, K., 2006. Introduction to Data Envelopment Analysis and Its Uses: With DEA Solver Software and References. Springer, New York.

Easton L., Murphy D. J., Pearson J. N., 2002. Purchasing Performance Evaluation: with Data Envelopment Analysis, European Journal of Purchasing & Supply Management,8 , 123-134.

Ertek, G., Can, M. A., and Ulus, F., 2007. Benchmarking the Turkish apparel retail industry through data envelopment analysis (DEA) and data visualization, EUROMA 2007, Ankara, Turkey.

Gattoufi, S., Oral, M., Reisman, A., 2004. A taxonomy for data envelopment analysis, Socio Economic Planning Sciences, vol.34, no.23, 141158.

Gattoufi, S., Oral, M., Kumar, A., Reisman, A., 2004. Content analysis of data envelopment analysis literature and its comparison with that of other OR/MS fields, J Opl Res Soc, vol.55, 911935.

Hoffman, P. E., Grinstein, G. G., 2002. A survey of visualizations for high-dimensional data mining, Chapter 2, Information visualization in data mining and knowledge discovery, Eds: Fayyad U, Grinstein, G. G. & Wierse, A., 47-82.

Hollingsworth, B., 1997. Review: A Review of Data Envelopment Analysis Software, The Economic Journal, vol.107, no.443, 1268-1270.

Keim, D., A., 2002. Information visualization and data mining, IEEE Transactions on Visualization and Computer Graphics, vol. 8, no.1, pp. 1-8.

Liu, J., Ding, F. Y., Lall, V., 2000. Using data envelopment analysis to compare suppliers for supplier selection and performance improvement, Supply Chain Management: An International Journal, vol. 5, no. 3, 143150.

Miner3D:
http://www.miner3d.com

Omniscope:
http://www.visokio.com/omniscope

Ross, A., Droge, C., 2002. An integrated benchmarking approach to distribution center performance using DEA modeling, Journal of Operations Management, vol.20, no.1, 19-44.

Smith, P.C., Goddard M., 2002. Performance Management and Operational Research: A Marriage Made in Heaven?, Journal of the Operational Research Society, vol.53, no.3, 247-255.

Spence, R., 2001. Information Visualization, ACM Press, Essex, England.

Soukup, T., Davidson, I., 2002. Visual Data Mining, New York, NY: John Wiley & Sons.

Sun, S., 2004. Assessing Joint Maintenance Shops in The Taiwanese Army Using Data Envelopment Analysis, Journal of Operations Management, vol.22, 233-245.

Talluri, S., Narasimhan, R. and Nair, A., 2006. Vendor performance with supply risk: a chance-constrained DEA approach, International Journal of Production Economics, vol.100, no. 2, 212-22.

Ulus, F., Kose, O., Ertek, G., Sen, S., 2006. Financial benchmarking of transportation companies in the New York Stock Exchange (NYSE) through data envelopment analysis (DEA) and visualization, 4th International Logistics and Supply Chain Congress, İzmir, Turkey.

Weber, C. A., 1996. A data envelopment analysis approach to measuring vendor performance, Supply Chain Management, vol.1, no.1, 2830.

Weber, C.A., Current, J.R., Desai, A., 2000. An optimization approach to determining the number of vendors to employ, Supply Chain Management: An International Journal, vol.2, no.5., 90-98.